FastReport v2.4

报表设计器

开发指南

翻译 ：张青松

E-mail: Z Q S@sohu.com

二00一年十二月

说 明

这是由小弟在去年年底开始翻译’大概花费了三个月的时间˚在翻译过程中’原文曾有多处不太理解’翻译不对之处’请各位大侠指正˚

本文仅供学习参考之用’不得用于任何商业用途˚

--------张青松

本书的著作权归作者所有˚由于张青松兄翻译的,但文档的存储方式采用的是 **Word** ˚文档大小竟然为 **8.37MB** ’太过于大˚为了各位朋友便于收藏与传播˚我将此文档制作成了 **PDF** 格式˚大小也只在 **1.1MB** 左右˚希望能给大家带来方便˚

---------罗祖术

2003 年 **6** 月 **6** 日

E-Mail:Luozs168@163.net

目 录

入门

绪言

这个指南让你熟悉报表设计器和了解报表设计的基本概念（各种区域，数据源，二次表，等等）。指南将帮助你开始用 FastReport 创建报表，但它不能告诉你怎样使用其它基本的报表设计器。

如果你不熟悉报表设计器，我们建议你参考 QuickReport 的帮助系统。QuickReport 的指南已经包含在你的 Delphi 拷贝中。QuickReport 的大部分基本概念也适用于 FastReport，然而，FastReport 能提供更多的弹性和最终用户的自定义。

关于 FastReport 的说明

FastReport 是高弹性的报表设计器，用于报表的数据可以从任何类型的数据源获取，包含字符列表，BDE 数据库，ADO 数据源（不使用 BDE），Interbase（使用 IBO），Pascal 数组和记录，以及一些不常用的数据源。

整个 FastReport 系统是用 Delphi 的 Pascal 编写的。FastReport 不需要动态链接库，但需在你的项目中占用大约 400kb（Delphi 5）。如果你想最终用户拥有设计能力，这将在你的.EXE 中增加大约 500kb。虽然这看上去比较大，但这只是其它设计的几分之一。你同样应该考虑到 FastReport 不仅仅只是包含最终用户更改报表设计的能力，还能够适应查询和数据库的变化。FastReport 还包含自己的脚本语言，让应用程序和最终用户能够更容易地改变报表。如果你的大部分应用使用 FastReport，你可以简单地配置 FastReport BPL（大约 1400kb）而所有你的应用程序只需要保留很少的一部分。

你可以发现 FastReport 有一个非常吸引人的用户界面，使用最新的用户界面组件，例如可停靠的工具栏。你的最终用户将会非常愿意使用这个设计器，只需使用鼠标就可以创建大多数报表。

FastReport 是名副其实的快速报表：较其它一些 Delphi 报表设计器而言，你可以发现没有什么可以接近于它的开发速度。报表预览窗口一直是大多数报表设计器的弱点，高品质的外观，赋于你的应用程序非常专业的用户界面。

FastReport 是一个已经拥有三年历史的非常成熟的报表设计器，成长使它拥有其它 Delphi 报表设计器所不能相比的诸多先进特性。

FastReport 的历史

FastReport 应运而生。当我在 1997 年开发一个工资系统时，我寻找周围的报表设计器能够让我非常简单地创建报表，而且能够在运行时编辑报表。可是，那时没有一个免费组件是我所需要的。这样，我就想创建自己的报表设计器，于是 FastReport 诞生了。

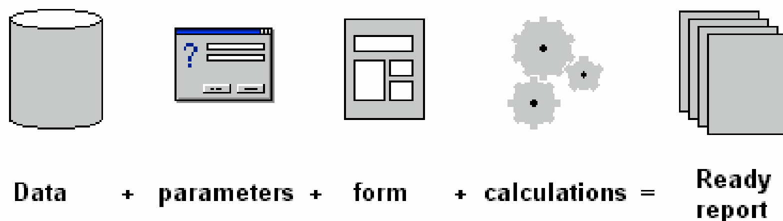
FastReport 的基本思想来源于 Windows 版的“1S-Bookkeeping”6.0，因此报表的基本元素是包含多行文本的矩形。这些文本既包含标准文本又包含变量。变量，像数据字段，用方括号突出显示。报表生成器的第一个版本仅支持一个区域，但它允许创建多层报表。它还不是一个组件，只是一些单元的集合。

后来，在 1998 年，报表生成器变成一个完整的 Delphi 组件。在这时，它命名为“FastReport”，并提供增强的函数功能。现在，FastReport 是一个完整的，产品化的可视化报表生成器。它包含以下的特点：

- 构建报表的生成器，也可以在运行时使用（重要的是用于最终用户报表的更改！）。
- 类似于 MS Word 的打印预览。
- 直接编译在 Delphi EXE 中，不需要动态链接库。
- 支持 JPEG（使用 Delphi 类库）和 GIF（使用 RX 类库）。
- 拥有比 QuickReport1 执行更快速的性能。
- 简洁，纯 Delphi 代码，比 QR3 有更小的设计器。
- 类似 QuickReport 和 ReportBuild 的强大的区域导向报表生成器。
- 包含许多非常有用的组件：文本，线段，图像，图形，OLE 对象，格式文本，RX Rich2.0，图表，图表，条形码，带阴影的文本。
- 报表生成的页数无限制。
- 多页报表；复合报表；子报表；分组；多栏报表；主从报表；透视表；二次表；
- 打印过程全控制；支持所有纸张尺寸；
- TXT，RTF（用于图形），CSV，HTML（用于图像）输出转换器。
- 在报表预览中查找文本。
- 附加的 TfrDataStorage 组件允许在运行时创建数据表和查询。这对于需要完全控制报告过程的最终用户报表特别有用。
- 报表生成的页面可以被修改。
- 内置的 Pascal 解释器可用于非常灵活的报告！语法加亮的支持是使用一个免费的软件库。
- 报表数据可以保存于 Delphi 的 DFM 文件，外部文件，数据表的 BLOB 字段或数据流。
- FastReport 可以非常容易地用于创建你自己的报表组件，向导和函数类库。
- 报表的核心部分支持 IBObject（不需要 BDE）。
- 支持 Interbase 表达式(IBX).
- 报表核心支持 ActiveX Data Object(ADO)。
- 完善的数据管理函数（不需要 BDE）。

创建报表

构建报表由下面几步构成：



你所看到的这些步骤将在 FastReport 中实现。同样我们将 FastReport 中的这些步骤与其它报表

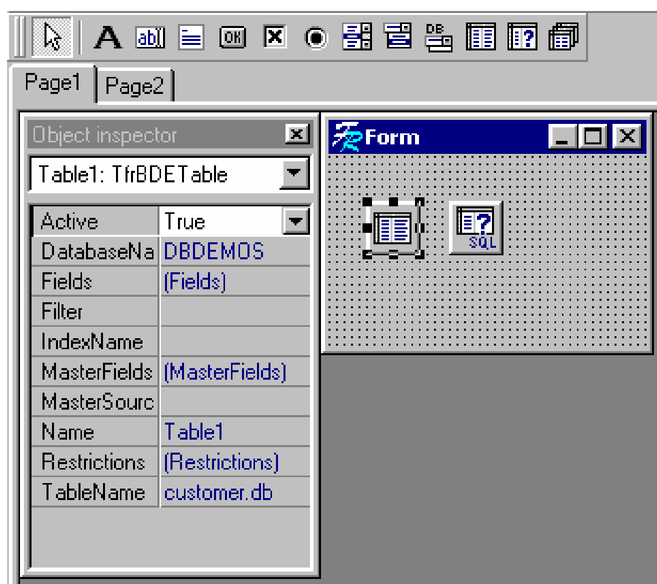
生成器相比较。

数据

大多数报表从数据库中查找数据。Delphi 本身提供了高效地数据库访问机制。这些机制也同样适用于 FastReport。TTable 和 TQuery 组件可以作为报表的数据源。总的来说它可以使用任何一个从 TdataSet 组件继承的子类。

除了数据，存储在数据库中，FastReport 事实上可以使用任何数据源（数组，文件，字符网格等等）。在这种情况下，程序必须自己控制非数据库源的存取。在 FastReport 中有一些事件集允许程序传递数据到 FastReport 的核心。

数据的存取与所有的报表生成器类似。所有的报表生成器都能处理放在项目窗体上的数据感知控件。除了存取数据，FastReport，ReportBuilder 和 QR+QRDesigner 都允许在运行时创建新的控件。在 FastReport 中创建的数据感知控件足以与 Delphi IDE 中的相提并论。同样，就象在 Delphi 中，你放置控件在窗体上，并在对象观察器中改变它的属性。控件的感知是非常灵活的：它能够简单地创建新的对象用于支持不同的数据引擎。

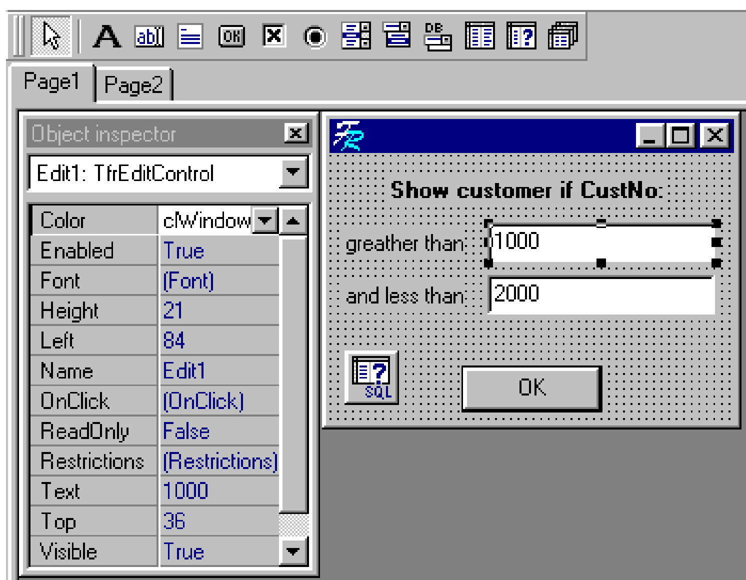


参数

用户可以输入参数。例如用户可以输入这些：数据范围，一个公司名称，一个发票编号，等等。少数报表不在所有的页中使用参数或使用固定的参数（不需要在对话框中取值）。

参数的处理在报表生成器中有不同的实现。在 ReportBuilder 和 QR+QRDesigner 中，如果报表使用查询中的数据，则有可能需要参数。为此将使用一个标准的对话框。除此之外，请求的参数还可用于在 Delphi 的 IDE 中创建的窗体。但请求的变化必须改变程序逻辑，并且需要重新编译你的项目。

FastReport 则相反，它允许最终用户开发对话框。这个过程就像在 Delphi 的 IDE 中一样：有一组标准控件，并能被拖放到对话框窗体上。如果需要，你也可以改变它们的属性。同样，FastReport 内置语言允许你使用伪 Pascal 脚本语言创建对话框，并输入变量到核心部分。

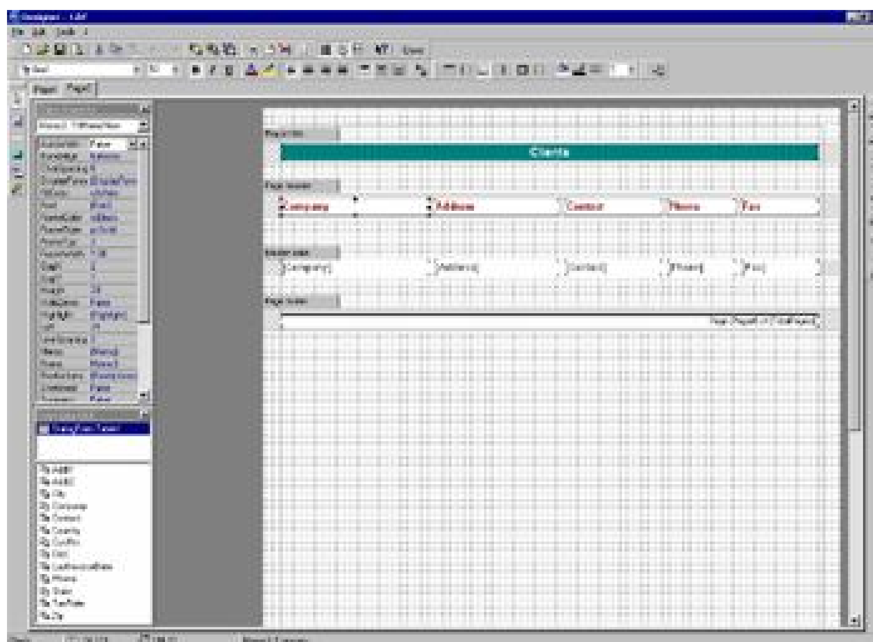


可能创建你自己的对话框是非常有用的。其它还可以允许你创建通用报表等等报表（创建数据源，使用内置语言，等等），而独立于应用程序之外（编译并创建项目）。这允许你不需要重新编写或重新编译项目就可能创建新的报表或修改已存在的报表。

窗体

报表窗体本身就是描述怎样可以看到正确的报表。为了在报表分组区域，FastReport 使用了“区域”。有两种类型的区域：服务器型区域（报表头，页码，等等）和用于报表的多行格式化文件部门的区域（未来：数据区域）。数据区域联接到数据源，内容是数据源中的记录。

报表开发的可视化环境——设计器——是用于构建报表的。在 FastReport 的设计器中你可以设计报表，强劲，简单又易于使用。设计器的界面由工具栏组成（Toolbars）。当然你可以改变它们的位置，只要你需要。使用对象观察器（Objects Inspector）可以控制对象的属性，类似于 Delphi 中使用的那一个。



处理

在构建报表的时候，处理输入的数据，报表窗体或者控件的修改。例如这样一个处理，用红色显示负数。一个复杂的处理是打印一个客户的产品汇总，在分组脚中进行处理（或分组头）。

实现这个处理是在 Delphi 中的事件处理中编写代码，并让它能够在 FastReport，QR 或 ReportBuilder 运行。这个方法并不通用，因为它不允许在 Delphi 外创建报表而无需重新编译。这就是为什么在 FastReport 中内置脚本语言，类似于 Pascal，但更简单。用这个脚本语言编写事件处理代码，在对象的处理过程中。这就可以让你创建复杂的处理而无需在 Delphi 中编写代码，并且，因此而不需要为项目联接一个报表。

Text editor

```
begin
  Cust := [CustomerData.RepQuery."CustNo"];
  if FinalPass then
    TotalSales := Arr[Cust] else
    TotalSales := 0;
end;
```

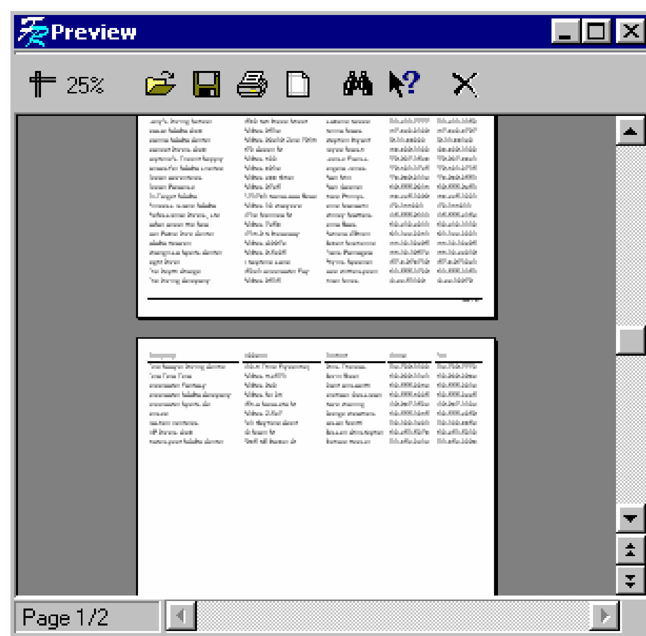
也可以用 FastReport 的脚本语言编写大规模的代码。在脚本中，你能使用所有对报表有用的属性和方法，以及变量和数据字段。在脚本中也同样可以创建变量和数组，当然，它们的值将可以用于所有的报表中。

准备报表

准备报表是激活 FastReport 的核心处理并能在点击“预览”按钮后进行预览的过程。与大多数报表生成器不同，它们的报表页面内容保存在图元文件（EMF 格式的图像）在 FastReport 中准备

好的报表是一组对象的集合，描述报表处理后各页面所包含的内容。这允许你可以修改生成的报表页面，通过在设计器中调用所需的页面。除此之外，还可以描述在预览窗口中鼠标点击对象的反应。这将让你更容易地组织你的工作（在一个报表对象点击可以产生一个更详细的新报表）。

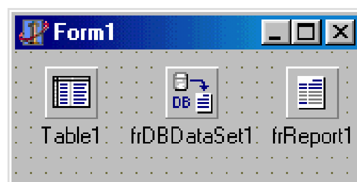
在 FastReport 的预览窗口中还有一点与其它报表生成器不同。在细节方面，FastReport 预览文档类似于 Microsoft Word：在一个窗口中可以立即看到多个页面。除此之外，你还可以在整个文档中查找文本。



举例：设计一个简单报表

使用 FastReport 构建一个简单报表。

1. 在一个窗体上放置三个控件：TTable（或一个 TQuery），TfrDBDataSet 和 TfrReport。你的窗体看起来像这样：



2. 联接 Ttable 控件到一个数据库（设置 DatabaseName 和 TableName 属性）。

3. 联接 TfrDBDataSet 控件的 DataSet 属性到 Ttable 对象。这个阶段窗体的 dfm 代码应该像这样：

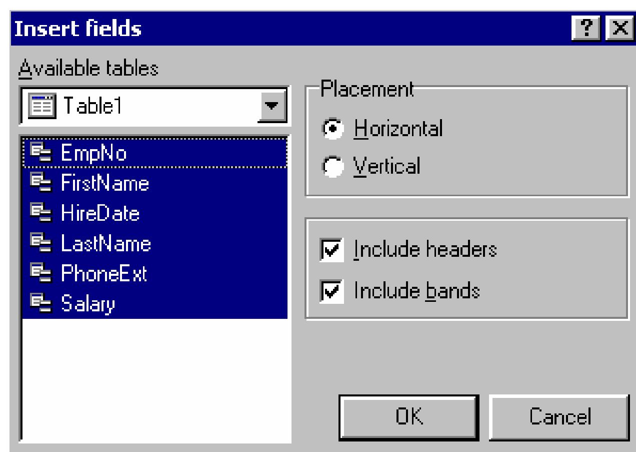
```
object Form1: TForm1
  Caption = 'Form1'
  object frReport1: TfrReport
```

```
      Left = 16
      Top = 8
    end
    object frDBDataSet1: TfrDBDataSet
      DataSet = Table1
      Left = 56
      Top = 8
    end
    object Table1: TTable
      Active = True
      DatabaseName = 'DBDEMOS'
      TableName = 'employee.db'
      Left = 96
      Top = 8
    end
  end
```

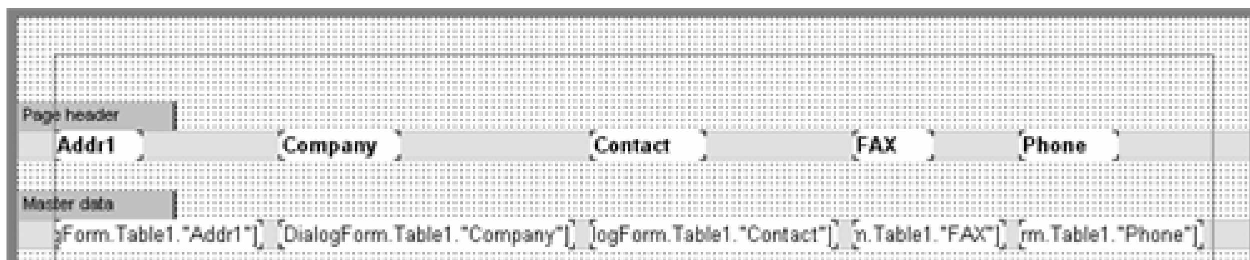
4. 打开报表设计器（双击 TfrReport 控件）。

5. 点击工具栏上的"插入数据字段" 按钮 。

6. 在对话框中选择所需字段并按"确认"键 。



现在报表包含你所选择的所有字段：



这可以自动生成一个报表模板，也可以包括图像，标题或页脚，页码，等等。

要运行报表，按工具栏的"预览"按钮，FastReport 现在构建这个报表并在预览窗口中显示内容。

Preview

60%

EmpNo	FirstName	HireDate	LastName	PhoneExt	Salary
2	Roberto	28.12.88	Nelson	250	40000
4	Bruce	28.12.88	Young	233	55500
5	Kim	06.02.89	Lambert	22	25000
8	Leslie	05.04.89	Johnson	410	25050
9	Phil	17.04.89	Forest	229	25050
11	K. J.	17.01.90	Weston	34	33292,9375
12	Terri	01.05.90	Lee	256	45332
14	Stewart	04.06.90	Hall	227	34482,625
15	Katherine	14.06.90	Young	231	24400
20	Chris	01.01.90	Papadopoulos	887	25050
24	Pete	12.09.90	Fisher	888	23040
28	Ann	01.02.91	Bennet	5	34482,8
29	Roger	18.02.91	De Souza	288	25500
34	Janet	21.03.91	Baldwin	2	23300
36	Roger	25.04.91	Reeves	6	33620
37	Willie	25.04.91	Stansbury	7	39224
44	Leslie	03.06.91	Phong	216	40350
45	Ashok	01.08.91	Ramanathan	209	33292,94
46	Walter	09.08.91	Steadman	210	19599
52	Carol	02.10.91	Nordstrom	420	4500
61	Luke	18.02.92	Leung	3	34500
65	Sue Anne	23.03.92	O'Brien	877	31275
71	Jennifer M.	15.04.92	Burbank	289	45332
72	Claudia	20.04.92	Sutherland		35699
83	Dana	01.08.92	Bishop	290	45000
85	Mary S.	01.08.92	MacDonald	477	35699
94	Randy	08.08.92	Williams	892	25900
105	Oliver H.	08.10.92	Bender	255	36799
107	Kevin	01.02.93	Cook	894	35500
109	Kelly	04.02.93	Brown	202	27000

Page 1/1

FastReport 内核

Delphi 控件面板

FastReport 对象

报表类型

Delphi 控件面板

安装 FastReport 后’将在 Delphi 控件面板中增加两页 ¬FastReport⌐页和¬FRTools⌐页°第一个页面包含 FastReport 的主要控件 例如 TfrReport TfrDesigher 等等 第二个页面包含 FastReport 中使用的附加控件’同样也可用于你的应用程序°

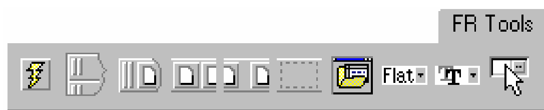
¬FastReport⌐ 页



图标	名称	描　　述
	TfrReport	这是主要的报表生成控件°用于编辑你的报表’在设计时双击这个控件打开报表设计器窗口°这个控件是所有报表都必需的°用于确定报表设计和版面°
	TfrCompositeReport	复合报表控件°这用于在单个报表中联合多个报表°这种类型的报表 你必须提供一组将被联在一起的 TfrReport 对象°这个控件仅在你需要构建复合报表时才需要°
	TfrDBDataSet	这个控件为报表提供数据源连接 这个数据源可以是 Ttable ︵通过 DataSet 属性︶或 TdataSource︵通过 DataSource 属性︶°这个控件是可选项’但在连接到一个数据库时是必需的°在报表中的每个数据区域必须有一个这个控件的实例°
	TfrUserDataSet	这个控件为报表提供数据源°与 TfrDBDataSet 所不同的是你必须通过事件提供数据项目°这个数据源允许你打印任何东西’例如一个数组’一个字符网格或一个文本文件°为这在数据源中进行记录导航’必须定义 OnFirst’OnNext 和 OnCheckEOF 事件°这个控件仅在你需要打印报表’但这个数据源不是所支持的数据库格式时才是必须的°在报表中的每个数据区域必须有一个这个控件的实例°
	TfrOLEObject	附加的 FastReport 对象°各个对象的功能将在下面讨论°如果你的报表使用了它们的一些功能’那么这个控件必须被拖放到窗体上°
	TfrRichObject	
	TfrCheckBoxObject	
	TfrShapeObject	
	TfrBarcodeObject	

图标	名称	描　　述
	TfrChartObject	
	TfrRoundRectObject	
	TfrTextExport	输出转换器。这些控用于将你的报表输出到一个所支持的外部文件格式（一般包括：Text，RTF，CSV 和 HTML）。仅在你需要输出功能才用到。
	TfrRTFExport	
	TfrCSVExport	
	TfrHTMExport	
	TfrDesigner	这是最终用户，运行期间的报表设计器。仅在你需要在运行期间设计报表才用到。
	TfrDataStorage	这个控件允许最终用户创建表和查询。这个控件已经淘汰了。用 TfrDialogControls 和 TfrBDEComponents 进行替代。仅在你需要给最终用户创建表和查询的能力时才用到。
	TfrPreview	当你想创建自己的预览窗口时可以使用这个控件。
	TfrPrintTable	这个控件直接打印表和查询的内容。
	TfrPrintGrid	这个控件直接打印 TDBGrid 所包含的内容。
	TfrDialogControls	这个附加控件包含了一组对话框控件，可以用于在运行时创建对话框。仅在你的报表中包含内置对话框时才用到。
	TfrBDEComponents	这个附加控件包含了一组数据库存取对象，例如 Ttable，TQuery 和 TdataBase。仅在你的报表包含内置数据库存取控件时才用到。

﹁FRTools﹂页



图标	名称	描　　述
	TfrSpeedButton	增强的 TspeedButton︒拥有﹁浮动﹂外观︵Flat 属性︶并且在鼠标移动到它的上面时可以选择颜色︵GrayedInactive 属性︶︒
	TfrDock	用于创建停靠工具栏︒
	TfrToolBar	一个模拟 MS Office 工具栏的工具栏控件︒
	TfrTBButton	工具栏按钮︒
	TfrTBSeparator	工具栏隔离带︒
	TfrTBPanel	包含其它 Delphi 控件的工具栏区域︐像 TcomboBox︐Tedit 等等︒
	TfrOpenDBDialog	﹁打开数据表﹂对话框︒显示一个模式对话框︐用于选择 BDE 别名︒
	TfrComboBox	浮动 TcomboBox︒
	TfrFontComboBox	包含已安装字体列表的浮动 TcomboBox︒
	TfrComboEdit	右边有一个自定义按钮的编辑框︒

让我们来浏览一下 FastReport 的主要控件︒



TfrReport 控件

这是 FastReport 的主控件︒它包含了调入︑保存︑预览和打印报表的方法︒每个 TfrReport 控件只能包含一个单独报表︒

TfrReport 属性

图标	名称	描　　述
DataSet	-	联接到一个 TfrxxxDataSet︒这个数据源的记录数定义需多少时间这个报表才能被构建和打印︒︵参考∶ReportType 属性︶
GrayedButtons	False	如果为 True︐设计器和预览窗口的工具栏将显示为灰色︒

图标	名称	描　　述
InitialZoom	pzDefault	定义预览的初始缩放值。
MDIPreview	False	显示预览窗口为一个 MDI 子窗口。
ModalPreview	True	如果为 True，预览窗口为模式方式。
ModifyPrepared	True	如果为 True，在预览窗口中双击页面则可以修改准备好的报表。
Preview	-	连接到一个 TfrPreview。如果这个属性被设置，准备好的报表将被显示在这个控件中。
PreviewButtons	All	定义在预览窗口中可用的按钮集。
ReportType	RtSimple	Defines how to interpret the data source connected to the DataSet property. If ReportType = rtMultiple, the report will be built as many times as the number of records in the connected data source. This is useful for printing a report multiple times based on a list (i.e. the data source). 定义怎样解释数据源连接到 DataSet 属性。如果 ReportType = rtMultiple，报表将在不同时间构建连接数据源的记录数。
ShowProgress	True	如果为 True，当准备、打印或输出报表时显示过度窗口。
StoreInDFM	False	如果为 True，存贮报表在 DFM 文件中。注意：报表存入 DFM 文件后存贮在可执行文件（EXE）中而不能被修改。
Title	-	报表标题。这个名称显示在预览窗口和打印任务列表中。

注意：TfrReport.StoreInDFM 属性缺省为 False。意思是你的报表必须存入一个可执行文件或转换到数据库的 BLOB 或二进制字段中。如果你想保存你的报表到 DFM 资源（像在其它的报表工具中，例如 QuickReport 和 ReportBuilder），设置这个属性为 True。但必需记住——这将要求你在任何一个报表被修改时都必须重新编译你的应用程序。

TfrReport 事件

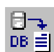
事件	描　述
OnBeforePrint	这个事件在打印报表之前被激活。*注意：FastReport 对象不是控件，因此它们不能显示在在对象观察中。你不能指定各个对象的个别事件处理。不过，你可以使用公共事件处理，像 OnBeforePrin，OnBeginBand 和 OnEndBand。*
OnBeginBand	在打印一个区域前这个事件被激活。
OnBeginColumn	在打印透视表栏前这个事件被激活。
OnBeginDoc	在打印一个报表前这个事件被激活。

OnBeginPage	在开始打印一个页面时这个事件被激活。
OnEndBand	在打印一个区域结束后这个事件被激活。
OnEndDoc	在结束一个报表的打印时这个事件被激活。
OnEndPage	在结束一个页面的打印时这个事件被激活。
OnGetValue	当 FastReport 在表达式中查找不明变量时这个事件被激活，指定事件处理返回这个变量值。
OnManualBuild	指定这个事件的处理将允许你手工构建报表（例如用代码显示一个需要的区域）。参考演示：DEMOS\MANUAL。
OnMouseOverObject	在预览窗口中当鼠标在任何时间移到某个对象时这个事件被激活。事件句柄将返回这个对象的光标类型（例如"pointing hand"光标）。这能让用户观察哪些对象可"点击"，哪些不能。
OnObjectClick	当你在预览窗口中点击某个对象时这个事件被激活。
OnPrintColumn	打印透视表栏前这个事件被激活。这个事件能够返回栏宽。
OnProgress	在长时间的运行中这个事件被激活。这个事件处理可以显示工作进度。
OnUserFunction	当 FastReport 在表达式中查找不名函数时这个事件被激活，这个事件处理将返回这个函数值。

TfrReport 的基本方法

方法	描 述
LoadFromFile, LoadFromStream, LoadFromBlobField	从一个外部文件，流或 BLOB 字段中调入报表。
SaveToFile, SaveToStream, SaveToBlobField	保存报表到一个外部文件，流或 BLOB 字段。
DesignReport	运行报表设计器。*你可以在你的应用程序中包含设计控件（TfrDesigner）就能使用这个函数。*

方法	描　述
ShowReport	构建报表并在预览窗口中显示它。*注意：在关闭预览窗口后，准备好的报表将被释放，因而不能保存到文件、输出或打印。*
PrepareReport	构建报表而不预览。*这个方法可以在调用 ShowPreparedReport，PrintPreparedReport，PrintPreparedReportDlg，ExportTo 或 SavePreparedReport 方法前调用。*
LoadPreparedReport	从一个外部文件中调用以前准备好的报表（.FRP 文件）。
SavePreparedReport	保存准备好的报表到一个外部文件。
ShowPreparedReport	显示一个准备好的报表。
PrintPreparedReport	打印一个准备好的报表。
PrintPreparedReportDlg	在打印一个准备好的报表前显示打印对话框。
ExportTo	使用输出转换器输出一个准备好的报表。

 TfrDBDataSet 控件

这个非可视控件 TfrDBDataSet 用于使用像 First，Next 和 Prior 方法遍历整个数据源。类似于 ReportBuilder 的 TppBDEPipeline 控件，但只能用导航而不能提供字段数据给予报表。

属性	描　述
CloseDataSource	在报表构建后关闭相关数据源。
DataSet	一个像 Ttable 或 Tquery 的数据集。
OpenDataSource	在构建报表前打开数据源。
RangeBegin	数据集中的首记录。
RangeEnd	数据集中的末记录。

这个控件的 DataSet 属性连接到一个像 Ttable 或 Tquery 这样的 TdataSet 控件。设置 OpenDataSource 和 CloseDataSource 为 True，用于自动打开或关闭相关数据集。

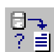
RangeBegin 和 RangeEnd 用于设置记录范围。

RangeBegin 可以接受下列值：

- rbFirst——从首记录开始选择记录；
- rbCurrent——从当前记录开始选择记录。

RangeEnd 可以接受下列值：

- reLast——在末记录停止记录选择；
- recurrent——当你开始报表时在当前记录处停止记录选择；
- recount——在选择 N 条记录后停止。记录选择数保存在 RangeEndCount 属性中。

 TfrUserDataSet 控件

这个控件也同样用于导航数据，但它用数据结构，像数组，字符网格，外部文件和以前的、已被替代的数据库，作为数据来源。

事件	描　述
OnCheckEOF	这个事件监视一个参数——EOF 的布尔值。设置这个参数为 True 可停止导航。
OnFirst	事件处理必须指定首记录指针。
OnNext	事件处理必须指定下一记录的指针。
OnPrior	事件处理必须指定前一记录的指针。

在这种情况下的"指针"和"记录"只是字面不同。如果你用来自数组的数据进行工作，"记录"是数组的行，而"指针"是保存行标的变量。

导航方法用下面的方式进行调用：OnFirst，OnCheckEOF，OnNext，OnCheckEOF，OnNext，…，OnCheckEOF，OnNext。如果是分组报表，那么 OnPrior 方法在分组结束后调用。

如果你在构建报表前知道"记录"数，你可以使用 RangeEnd 和 RangeEndCount 属性。只须设置 RangeEnd := reCount 和 RangeEndCount :=记录数（例如，数组的行数）。这时，你可以不需要 OnCheckEOF 事件处理。要找到当前记录位置，使用 RecNo: Integer 属性。在第一个位置 RecNo = 0。

FastReport 对象

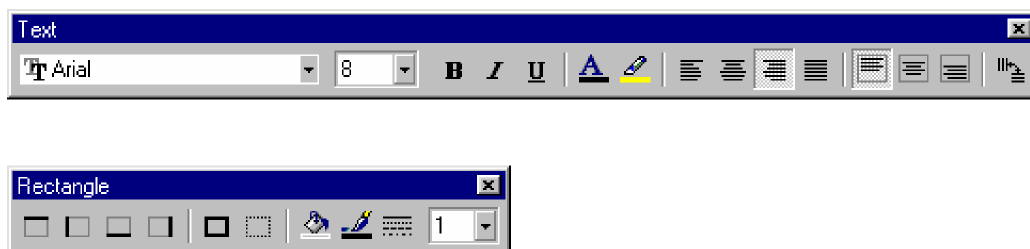


图标	名称	描　述
	Text	提供里面包含多行文本的矩形框。文本允许包含变量。
	Band	条状区域。定义区域包含的最终报表的位置。
	Picture	显示 BMP、ICO、WMF、EMF 和 JPG 图像格式，来源可以是 BLOB 字段。不连接到 FR.inc 就不能在 FastReport 中使用 JPG 格式。
	SubReport	用于创建子报表。当你插入这个对象到一个报表时，你将看到在你的报表中增加一个新页面。
	Line	在报表上画垂直或水平线条。
	Shadowed text	提供包含在一个矩形边框中的多行文本，并可设置阴影和（或者）渐变色填充。用于打印标签。
	Barcode	用条形码显示数据。
	Shape	允许在报表中插入几何图形（矩形，圆角矩形，椭圆，三角形）。
	CheckBox	用检查框或 X 来显示逻辑数据。
	RichText	用于在报表中插入 RTF 文档。可用 BLOB 字段作为数据源。
	OLE	用于在报表中插入 OLE 对象。
	Chart	用于在报表中插入图表。TeeChart 控件必须在 Delphi 中安装后才能正常工作。
	RichText 2.0	类似 RichText 控件，但允许图像和 OLE 对象插入 RTF 文本中可用 BLOB 字段作为数据源。 TrxRichEdit 控件和 RX 类库在 Delphi 中已经安装才能使用这个控件。在 FastReport 中激活不需要在 FR.inc 中加入相关行。



“Text”对象

这个对象无疑是 FastReport 中最强大、最灵活的对象。主要是，它提供一个包含多行文本的矩形框。你可以设置边框的类型、颜色和宽度；以及字体属性、文本对齐方式和字体方向（垂直或水平）。要设置对象的属性，使用“文本”和“边框”工具栏：



text 对象的内容主要由备注类型的对象组成，可以包含：文本，变量，数据字段，或这些的任意组合。字体格式将应用于 text 对象所包含的所有文本。

使用 text 对象的一些示例：(*)

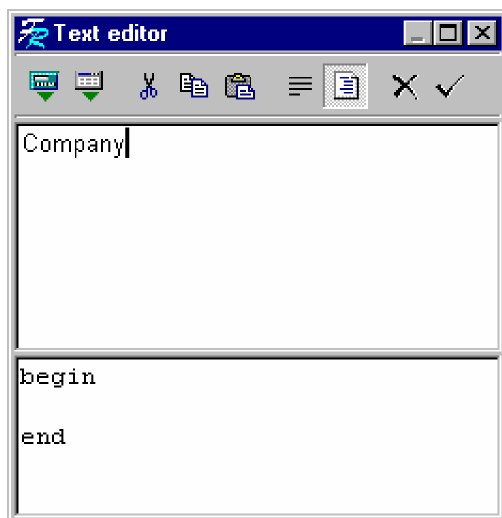
宽度，cm：[Length] | | 使用变量（[Length]）显示固定文本（宽度，CM）。

宽度，cm：[Table1."Length_cm"] / / 使用数据字段（[Table1."Length_cm"]）显示固定文本（宽度，CM）。

*宽度，cm：[[Length inch]*2.54]* | | 使用变量表达式（[[variable]*value]）显示固定文本（宽度，CM）。It is important to note the additional set of square brackets when using formulas with variables.

宽度，cm：[Length] | | 使用数据字段表达式（[datafield*value]）显示固定文本（宽度，CM）。Notice the single set of square brackets is required for data fields and formulas.

修改 text 对象最快速的方法是在报表编辑设计器中选择矩形（点击它）并双击。这将出现文本编辑器对话框：



这儿有工具栏按钮的一些简短提示：

插入一个表达式；

插入一个数据字段；

剪贴板操作；

自动换行选项；

脚本编辑选项；

取消按钮；

确认按钮。

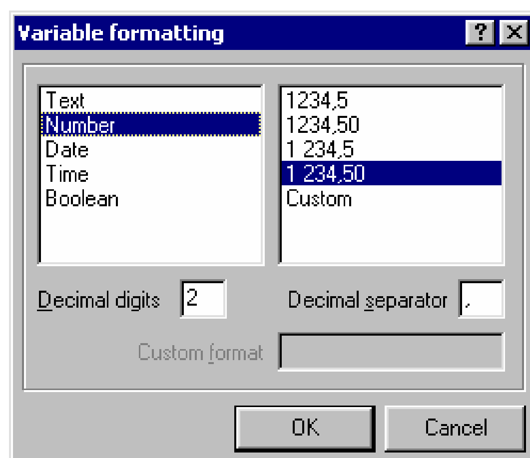
编辑器中你可使用的文本类型，可插入变量、数据字段或表达式。点击编辑器中的“数据字段”或“表达式”按钮将显示常用的数据字段或变量。你同样也可以使用下列键：

Insert　　“变量”按钮——显示变量对话框。

Ctrl+Enter　　“确认”按钮——确认并关闭对话框。

Esc　　“取消”按钮——关闭并取消编辑。

每个 text 对象允许保存自己的格式。你可以在对象编辑器中点击 text 对象的“DisplayFormat”属性来编辑格式。格式选项包括：无格式，显示为一个数值、日期、时间或逻辑格式。



对于各种格式类别 ’你可以选择预设格式中的一个 ’或者自定义格式︵例如 ’#,##0.000 表示一个数值︶°格式化使用与 Delphi 相同的方法︵这个描述在 Delphi 的在线帮助中 ’参考﹃Formatting Strings﹄主题︶°逻辑值被允许使用下列格式 **False_string ；True_string** °︵例如要显示﹃是﹄或﹃否﹄’使用﹃是 ；否﹄作为格式︶

格式可以应用于文本中的每个或全部变量 °如果一个变量不能被格式化 ’它将作为一个纯文本显示 °如果你在一个 text 对象中使用多个变量 ’但你想为每个变量单独格式化 ’你可以不用缺省而使用«#»标记格式化 °放置这些标记和格式字符在变量的后面 ’等等 ：

[Variable #format] ’这儿的格式为下列之一 ：

- x.x 或 Nx.x 或 Nyyyyy｜｜数值格式 °x.x ｜｜整数/小数部分的宽度 ；yyyyy ｜｜类似#,##0.00 的字符 (描述在 Delphi 的在线帮助系统中 ﹃Formatting strings﹄主题) ° 如果 x.x 或 yyyyy 字符串包含«.» `«,» `«-» 字符 ’这些字符将用于小数点或分隔符 °
- Dxxxxx, Txxxxx｜｜日期和时间 °Xxxxx｜｜类似于 dd.mm.yy 的字符串 °
- Bxxxxx;yyyyy｜｜逻辑格式 °如果变量为 False ’则显示 xxxxx 字符 ；否则 ’显示 yyyyy 字符 °

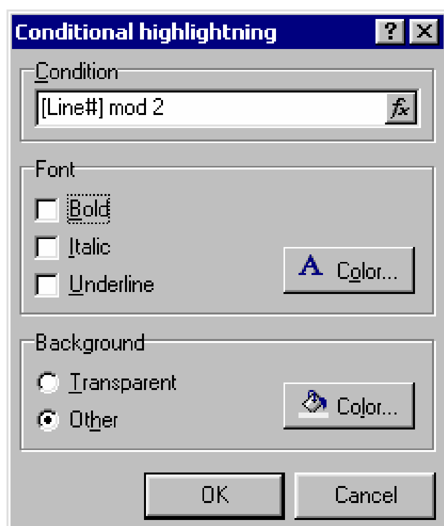
这儿有一些使用«#»标记的示例 ：

[Table1.«N1» #9.2] [Table1.«N2» #N9-2] [Table1.«N3» #N#,##0.00]｜｜数值格式 [Table1.«Date1» #Ddd.mm.yyyy] [Table1.«Time» #Thh:mm:ss]｜｜日期/时间格式

[Table1.«Bool1» #BFalse;True] [Table1.«Bool2» #BNo;Yes]｜｜逻辑格式

你不能在变量编辑中创建的表达式使用格式标记︵见下面︶°

对于报表你希望根据条件或表达式改变对象的字体颜色 ’背景色等等 ’在文本格式工具栏中点击按钮 °这将出现﹃条件加亮﹄对话框 ’你可在编辑框中输入条件和设置不同的选项 °



例如，要挑选出超过$1,000.00 的订单，示例报表“3-level”就是一个这样的例子。要做到这一点，选择包含订单合计的矩形框并选择格式工具栏中的高亮按钮。在编辑框中输入条件“Value > 1000”。选择一个适当的字体和背景颜色，并点击确认按钮。你的报表会高亮显示超过$1000.00 的订单合计。试一下你所需要的不同字体和背景的组合。

在“Text”对象上右击，你可设置下列选项：

- 伸展——对象的高度将根据变量的实际高度进行调整。你同样也可以设置对象所在区域的这个选项。当这个区域被打印时，它将根据所有伸展对象和它自己计算最大高度。
- 自动折行——长字符串自动折行为多行文本。
- 自动宽度——在画之前，对象才计算它的实际宽度。
- 仅文本——不处理对象中的变量。
- 禁止重复值——不显示重复的值。

同样你还可以找到对象观察器中的一些附加属性：

- CharSpacing——字符间距；
- GapX, GapY——左边和顶部间隔;
- LineSpacing——行间距。

“Band”对象

像其它可视化报表设计器，FastReport 是一个基于区域的报表生成器。意思是你可以在报表上放置区域，然后在区域放置数据。下表列出当前 FastReport 支持的区域类型，以及在最终报表中它们的位置。

注意：区域并不是按照报表设计器中的顺序出现。*区域类型*决定将它们在最终报表中的位置。然而，放置区域在恰当的位置会使它更于修改。

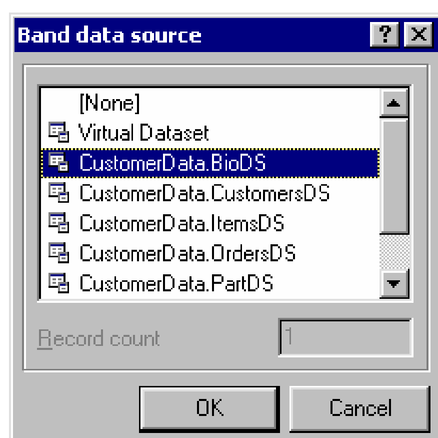
名　称	何　时　何　地
报表标题	打印在打印报表之初
报表合计	打印在报表结束时
页头	打印在每页项部
页脚	打印在每页底部
主项头	打印在第一个数据层开始
主项数据	第一层数据丨丨重复每个主项数据记录
主项脚	打印在第一个数据层结束
细项头	打印在第二个数据开始
细项数据	第二层数据丨丨重复每个细项数据记录
细项脚	打印在第二个数据层结束
子细项头	打印在第三层数据开始
子细项数据	第三层数据丨丨重复每个子细项数据记录
子细项脚	打印在第三个数据层结束
覆盖	打印在每页的最底层︵用于打印水印︶
栏目头	打印在栏目开始
栏目脚	打印在栏目结束
分组头	组标题打印在分组开始
分组脚	打印在分组后
交叉表头 交叉表数据 交叉表脚	这组区域用于创建交叉表 ’报表上可变栏目 。
子	这个区域能被追加到其它任何类型的区域︵交叉表和页脚除外︶。

	子区域显示在父区域之后。

正如你所看到的，这组区域与其它普通接受的不同。在正统的方案中（在细节，在 ReportBuilder 中）只有一个细节数据区。

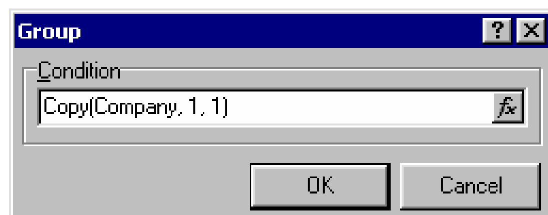
依靠区域类型，FastReport 展示了合适的编辑器。

对于从一个数据源显示数据的区域（细项数据区域，主项数据区域，等等），你将被提问区域从何获取数据。你将可以显示一个数据集的所有变量或你可以选择"虚拟"数据项。



当你选择虚拟数据集选项，你必须指定包含在"数据数据集"的"记录"数。如果指定的记录实际存在，区域可以正常工作。虚拟数据集能帮你打印窗体，多条线需要被打印，但你不希望手工一个一个地插入。你可以简单地设计一个单线，然后使用虚拟数据集选择出现的次数。

如果你选择分组头区域类型，你将要求编辑分组头。这个编辑器允许你定义分组条件，基于数据表的字段或其它一些表达式（例如按姓名的第一个字母进行分组，同样也能用于电话目录）。



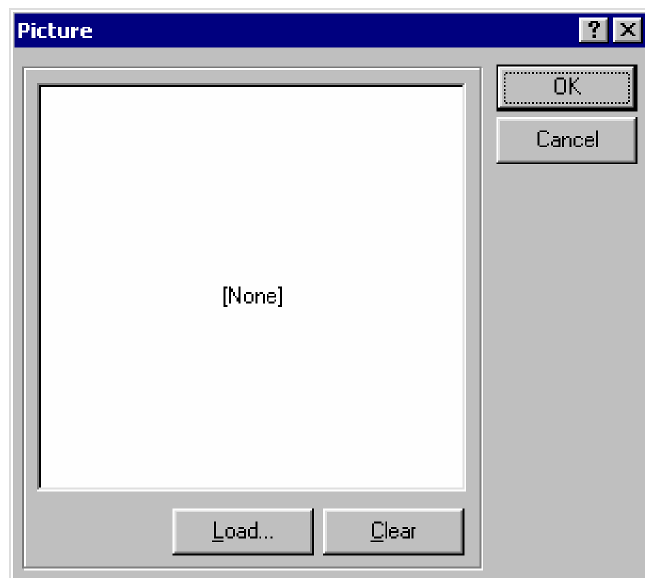
如果你在区域上右击，你将看到一组快捷菜单，包含：

- 伸展——这个意思是由区域中的对象高度决定区域高度（"Text"对象的自动折行选项激活时将按需要伸展。这允许自动调节区域（行）高度以适应行中的所有数据。
- 分隔——在页面未满之前报表新页面。
- 强迫换页——强迫区域内容打印在一个新页面。
- 在第一页——这个区域将打印在第一页（仅适用于页头和页脚）。

- 在末页——这个区域将打印在末页（仅适用于页脚）。
- 在所有页重复——这个选项仅用于主项头、细项头、子细项头、分组头和交叉表头区域。如果这些头下面的数据强制到一个新页，这些头也将出现在那些页面中。

“Image”对象

图片对象用于插入图像到一个报表中。图像格式支持包括 BMP/WMF/ICO。JPG 和 GIF 也可以插入，但不直接支持，它们需要添加 Delphi 图像支持类。这些类可以从其它地方免费得到。



图片对象编辑器用于选择一个图像，清除图像或从数据库的一个 BLOB 字段选择一个图像。需要从文件插入一个文件，双击图片对象。需要从 BLOB 字段手稿图像，选择图片对象，按 Ctrl+Enter 调用文本编辑器，然后在文本编辑器中调用 BLOB 字段名，例如：[Tabel1.GraphicField]。这个操作执行从文本编辑器中插入的数据字段选择图片。

如果要直接打印一个数据库中的加密图像，使用 OnBeforePrint 事件在报表运行时解密图像并调用到运行时的报表中。在手册的后面可以找到一个使用 ADO 和 SQL 服务器的例子。

如果要被打印的图像存贮在磁盘中，使用 TfrReport.OnBeforePrint 在打印之前调用图像。一个打印指定目录中图标的例子可以在手册的最后找到。

图片对象的快捷菜单允许设置下列选项：

- 伸展——图像将伸展到整个矩形区。
- 锁定纵横比——如果缩放则图像的比例是受保护的。
- 居中——图像在矩形区中居中。
- Blob 类型——BLOb 流类型：BMP, WMF, ICO or JPG。因为 Delphi 不支持流类型自动识别，所以这是必需的。使用对象观察器设置这个属性。

 “SubReport”对象

Subreport 对象作为在一个报表中特定位置插入附加报表的占位符。Subreport 中的报表打印必须在主报表中的单独页。一个使用 Subreport 的例子是插入一个图表在下方或紧跟着一些销售数据。

当插入一个 Subreport 对象到报表，一个附加页自动插入到主报表中。Subreport 自动指向新插入的页面。Subreport 中任何附加到新页 Anything added to the new page will be printed within the Subreport object instead of being shown on the main report.

Subreport 对象将被放在主报表下方。如果多个 Subreport 对象将要被放在各自下方，插入它们到单独的数据区域。

当使用 Subreport 时有一些限制：

- 不能使用分栏；
- 下列区域能用于 Subreport（但不能关联到主报表）：报表标题，报表合计，页头，页脚，栏 XXX 区域；
- 不能使用区域暂停；
- Subreport 中不能使用分组。

 “Line”对象

直线对象用于插入水平线或垂直线到报表。当在报表中分隔细节区，使用直线可以更容易阅读。直线的粗细和颜色可以使用绘图工具栏调整。

要画一条线，单击直线工具栏按钮，在当前页面上拖动鼠标，光标将变成一支铅笔画出一根直线。单击鼠标的地方就是直线的起始端。按下并不松开鼠标，然后移动鼠标到直线的末端。松开鼠标后，直线就画好了。直线可以被选中并按需要修改。

 “CheckBox”对象

检查框对象是用于图形显示逻辑数据，并只能联接到逻辑变量或数据字段。当变量或字段值为 True，则显示一个叉。如果值为 False，则为空。

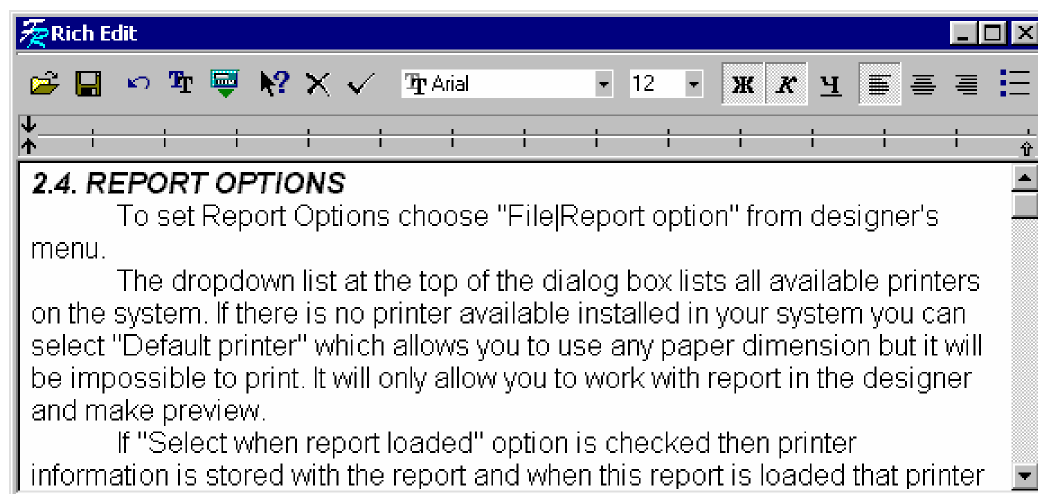
注意：如果一个检查框对象插入到报表，TfrCheckBoxObject 组件必须被插入到同一个窗体中或者在运行时产生一个错误消息框。

 “RichText”对象

RichText 对象用于打印 RTF（Rich Text format）文档。当在报表中需要格式化文本而文本对象又不能满足要求时可以使用 RichText 对象。

RichText 能提供 FastReport 的 TextObject 所不能提供的对文本格式的高度控制。RTF 文件可以使用类似 MS Word 的程序创建或使用内建的 RTF 编辑器，它基于 Delphi 的 RTF 编辑器。RichText

对象将保留和显示 RTF 文件中的所有格式。



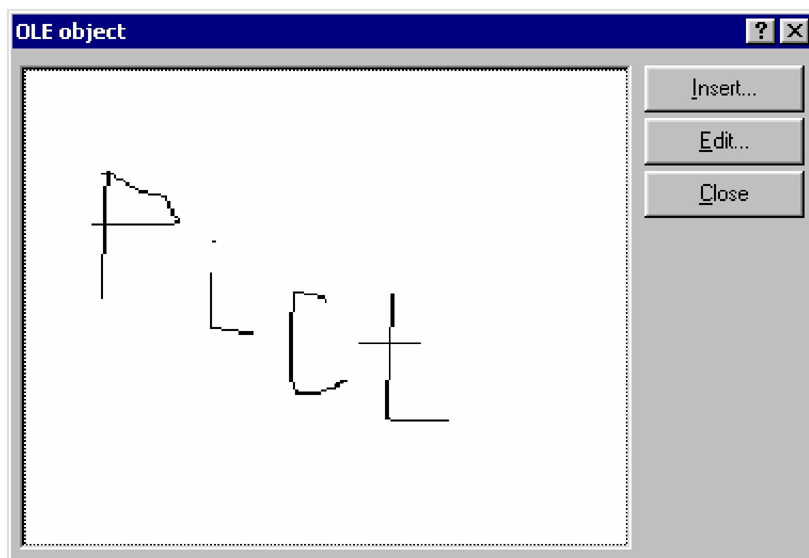
FastReport 所包含的 RTF 编辑器支持文本的 RTF 所有基本选项 变量能被插入到格式化文本中。这个变量必须加上方括号（就像在文本对象中一样）。

RichText 对象也能用于打印数据库 BLOB 字段所包含的格式化文本。要选择所需字段，按 Ctrl+Enter 调用文本编辑器或者直接输入方括号中的数据表字段，例如：[Table1."RichField"]，或单击“插入数据字段”并选择所需的数据源和字段。

注意：一个 TfrRichObject 对象必须被手稿到同一个窗体中。如果不这样，则会产生一个错误消息。

 “OLE”对象

OLE 对象用于插入一个 OLE 对象到报表中。OLE 对象编辑器允许插入一个新 OLE 对象。单编辑上的插入，则调用标准的 OLE 插入对象对话框用于显示所有可用的 OLE 对象。



OLE 对象能被用于利用 OLE 对象包含一个 BLOB 字段。要达到这个要求，按 Ctrl+Enter 打开 OLE 对象观察框，选择插入，查找所需 OLE 对象。

注意 1：在快捷菜单中的扩展选项有时可以观察 Excel 数据。

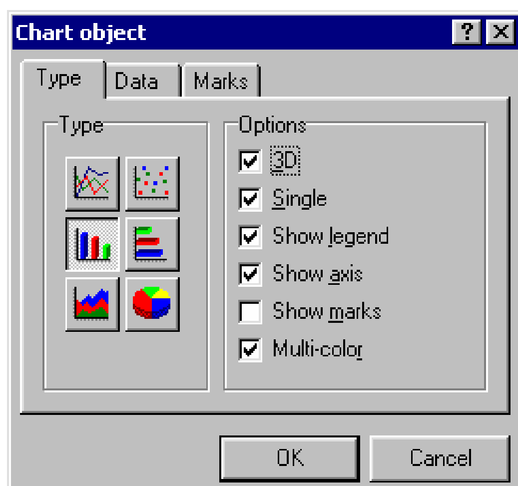
注意 2：一个 TfrOLEObject 对象必须被手稿到同一个窗体中，否则将会在运行时产生一个错误消息。

 “Chart”对象

Chart 对象用于在报表中插入图表，特别可用于销售图表的打印。

图表编辑器的类型页可以选择六种类型的图表以及下列选项：

- 3D－显示 3 维图表；
- 单线－
- 显示图例：在图表的附近显示图例；
- 显示轴：显示轴（不能用于饼图）；
- 显示标记：显示标记；
- 多色显示：使用不同的颜色显示图表变量。



要连接图表到数据字段’你需要设置两个 Text 对象’即数据字段’并已在报表上°这些数据字段的内容将用于设置图表的变量和图例°图例通常显示在 X 轴上’变量在图表的 Y 轴上°当生成报表时’选择的 Text 对象的内容将保存到图表对象的备注中°右击 Chart 对象’选择﹃编辑﹄中的﹃数据﹄’然后设置﹃图例﹄和﹃值﹄’图例设为﹃memo2﹄’值设为﹃memo6﹄︵细节请参考示例︶°

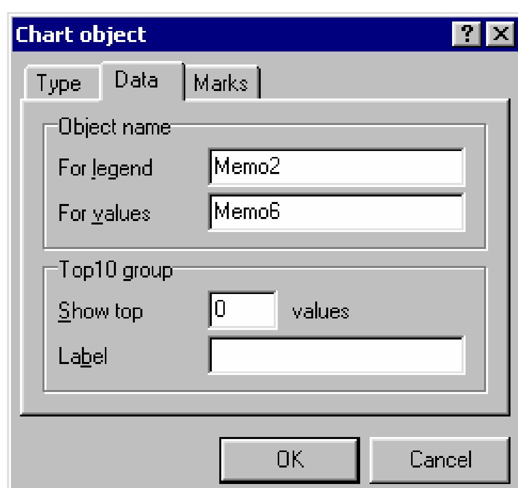
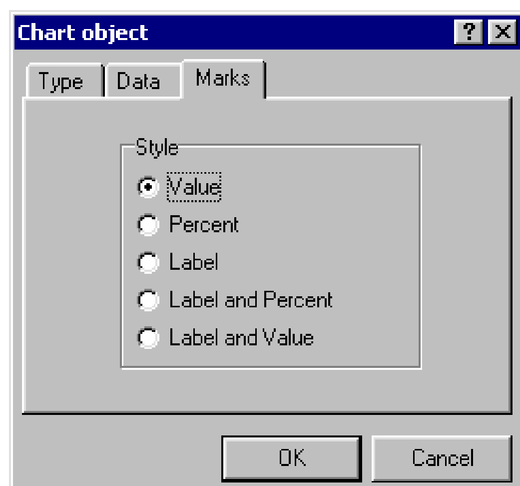


Chart 对象允许创建﹃前 10 个﹄图表°这将在报表中仅包含最大的几个值’而排除其它值°要做到这一点’在图表编辑器的数据页中设置前几个值以及其它未包含值的图例︵通常用﹃其它﹄表示︶°

图表编辑器的标记页允许选择标记的类型︵﹃值﹄是缺省的︶°如果要显示选中的标记就必须在选项中选择﹃显示标记﹄选项°



如果用于显示图表值的 Text 对象包含的是有格式的变量（例如，10000.00 或$100.00），则 Chart 对象尝试从这个字符中查找数据。如果没有在字符串中找到数据，那么将忽略所有的字符。如果有更多高级格式被设置（例如，10000km2），那么这个值就不能用于 Chart 值。为这些变量创建无格式但不显示的对象，就可作为图表中的值。只需在对象编辑器中设置 Visible 属性为 False 就可隐藏对象。

图表的值是累积保存在 Chart 对象的 memo 中：

Header1;Header2;Header3

Value1;Value2;Value3.

一个 Chart 允许使用 Text 编辑器（Ctrl+Enter）编辑。在 Momo 对象中插入适当的值。

注意 1：TeeChart 控件包含在 Delpi3 及以上的版本中。要用于 Delphi2，必须先安装 TeeChart 控件，然后正确配置 FR.INI 文件并重新编译 FR 类库。

注意 2：一个 TfrChartObject 控件必须放在窗体上。否则，将会在运行时产生一个错误消息。

 “Shape”对象

Shape 对象用于在报表中插入几何图形（矩形，圆角矩形，椭圆，三角形）。

注意 1：当使用三角形时，背景和填充色缺省为白色，并不能更改。

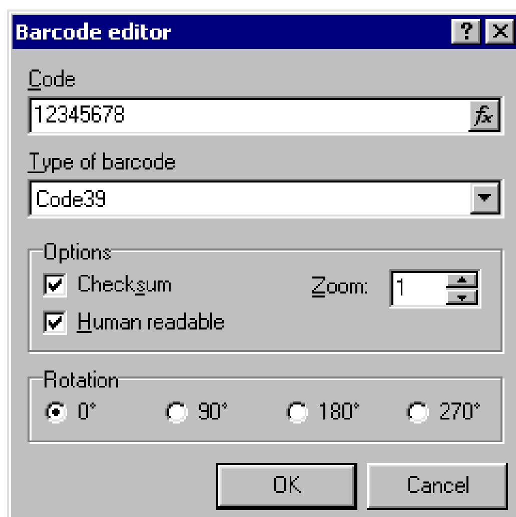
注意 2：一个 TfrShapObject 控件必须放在窗体上。否则，运行时将会出错误信息。

 “Barcode”对象

Barcode 对象用于打印条形码。条形码的宽度由数据的大小决定。下列条形码格式已被支持：

- 2 of 5 interleaved

- Code39
- Code39 Extended
- Code128A-C
- Code93
- Code93 Extended
- MSI
- PostNet
- Codebar
- EAN8
- EAN13
- EAN128A-C
- UPC A, E0, E1, Supp2, Supp5



提示：如果条形码四周需要一些空白区，设置 **FrameTyp** 属性为 15，这将在整个对象四周画一个连框，**FrameWidth** 设为 6.00，**FrameColor** 设为 clWhite。这可以完成类似于工具栏按钮的连框。

注意：一个 TfrBarcodeObject 控件必须放在窗体上，否则，在运行时将出现错误信息。

 “RichText 2.0”对象

RichText 2.0 对象除了它是基于 RX Rich 控件以外，与 RichText 对象一样。它允许在 RTF 文本中包含 OLE 对象和图像。如果 RX 类库已安装（必需为 RXLib 2.0 或更高版本）就可以使用。要激活这个对象，从 FR.INC 文件中移动{$DEFINE RX}这一行的注释并重新编译 FR 包。

注意：一个 TfrRxRichObject 控件必须放在窗体上，否则，运行时将出现错误信息。

创建报表

要开始报表，使用 FR 控件面板上的 TfrReport 。它能够存储一张报表，打开或保存它，运行，预览和打印。在设计时双击这个控件，你就可以看到报表设计器。

大部分报表都包含数据库中的数据。要访问数据，FR 使用标准的 Delphi 数据感知控件 TTable 和 TQuery 。当然，你也可以使用"任何"其它基于 TdataSet 和 Tfield 的控件。

FR，与其它报表生成器一样，使用 bands 逻辑组织报表数据。Bands 分成两类：data-bands（主数据，从数据，等等）和其它（报表标题，页头，等等）。Data-bands 出现在报表的多行部分，例如 Ttable 的记录行。你可以指定一个数据集到这样的 band。FastReport 有两个用在这里的控件：FR 控件面板中的 TfrDBDataset 和 TfrUserDataset 。事实上 TfrDBDataset 控件用于连接象 Ttable 这样的数据感知控件和 data-band。TfrUserDataset 适用于当你处理的象数组、网格等这样的非数据库数据。

好了，在创建报表之前，你必须做到：

1)放 TfrReport 控件到窗体上；

2)如果你的报表使用数据库数据，放置象 TTable, Tquery 的数据感知控件到窗体上；

3)对于你的报表上的各个 data-band 你必须创建管道(TfrDBDataset 或 TfrUserDataset)并设置它们的属性。

FastReport 中的 Bands

FastReport 有 22 个 Bands，列表如下：

名　　称	何　时　何　地
报表标题	在报表开头打印
报表合计	在报表结尾打印
页头	在每页顶部打印
页脚	在每页底部打印
主项头	在主数据开头打印
主项数据	主数据，重复主数据记录
主项脚	在主数据结尾打印

细项头	在细数据项开头打印
细项数据	细数据项ı重复细数据记录
细项脚	在细数据项结尾打印
子细项头	在子细数据项开头打印
子细项数据	子细数据项ı重复细数据记录
子细项脚	在子细数据项结尾打印
覆盖	在每页的最底层打印︵用于打印水印︶
栏目头	在栏目的开头打印
栏目脚	在栏目结尾打印
分组头	在分组打印分组标题
分组脚	在分组后打印
交叉表头 交叉表数据 交叉表脚	这组 bands 用于设计交叉报表’一种可变栏目报表°
子	这个 band 能添加任何类型的 band︵例如交叉表头和页脚︶°子 band 在父 band 后显示°

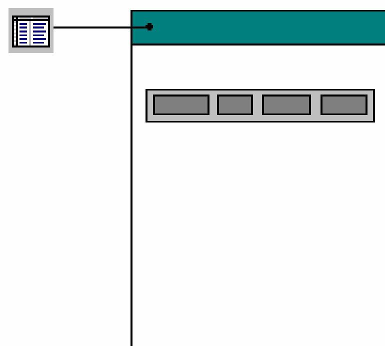
正如你所见’这组 bands 集与 ReportBuilder︵现在和以后的版本ı ReportBuilder 4.X︶中的﹃一流﹄方案不一样°在这个方案中’报表自身包含一个数据集和它出现的主项数据°细项数据在细项数据 band 中定义格式°换句话说’这个方案允许你创建主/从报表°如果你需要更多的子细项数据’你被迫使用 sub-report’事实上在附加的页上放置其它报表°Sub-report 用于创建主ı从ı从报表°

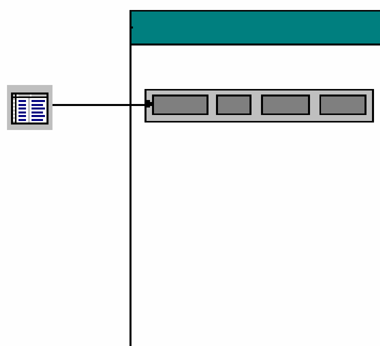
FastReport 中的方案’你不需要 sub-report°在这个表中你可以看到’你可以在一个报表中使用三层数据︵bands∶主项数据’细项数据和子细项数据︶°每个数据 band 显示一层数据’但必须有一个管道 TfrDBDataset 或 TfrUserDataset 指向它°如果你需要更多的层’你可以使用子(subreports)°

你可以在一个报表中放多个数据 band°例如’你可以放两个主项数据在同一个报表中ı它允许你简单地创建﹃一对一﹄报表°

为了证明上面的结论’我们分别用 FastReport 和 ReportBuilder 来创建一个报表°

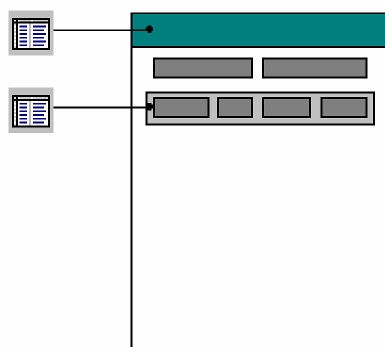
例一∶单层数据报表

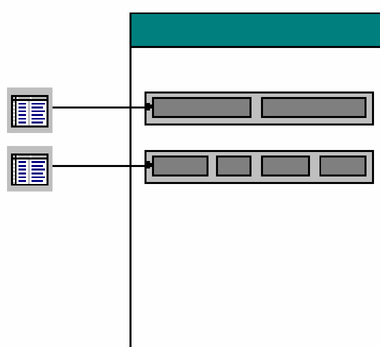




ReportBuilder FastReport

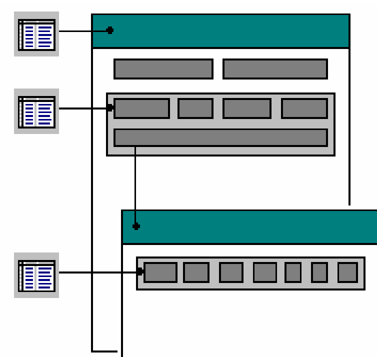
例二 ：两层数据报表

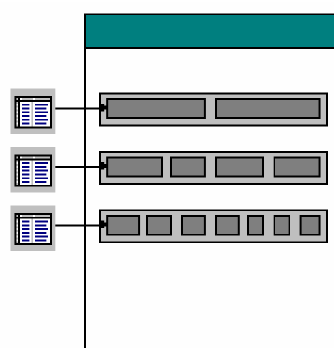




ReportBuilder FastReport

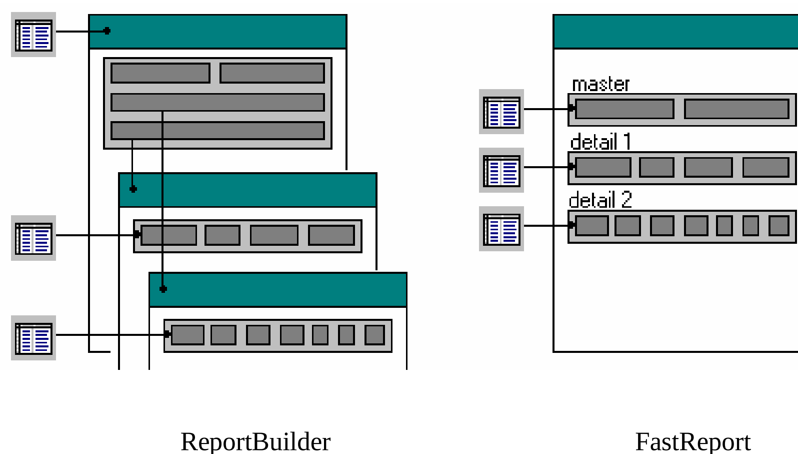
例三 ：三层数据报表





ReportBuilder FastReport

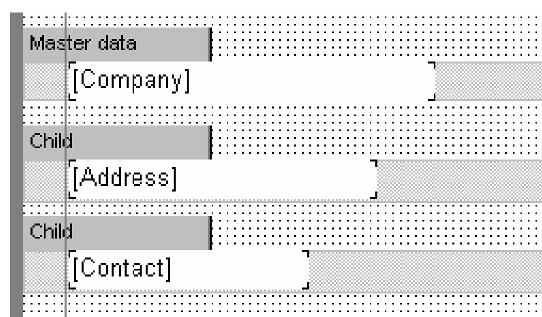
例四 ：主 | 从 | 从—报表



ReportBuilder FastReport

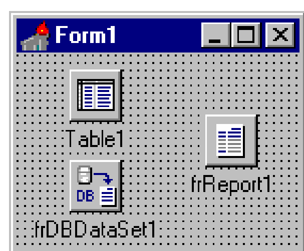
FastReport 中使用的方案在报表组织中有更多的弹性，正如我们所看到的，FastReport 允许指定到对象"Report"的数据源。TfrReport 对象有些许多相关的我特性：ReportType（rtSimple，rtMultiple）和 DataSet：TfrDataSet。如果 ReportType=rtSimple（缺省），你可以不指定数据源到 band 主项数据，它将调用 TfrReport 对象的 Dataset 属性。否则（ReportType=rtMultiple）the report would be serially built so many times, as the number of notes in the data set, connected up to the DataSet feature.这将便于打印报表的多份拷贝，但必须设置拷贝份数（e.g. to write down on the first copy "Primary" and "For acquaintance" on the rest copies）。

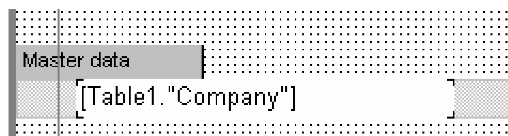
有两个辅助 bands 可以附加在主 bands 上，在一流方案中不包括：覆盖和子。这些 bands 是从 QuickReport 借鉴来的。"覆盖"用于打印背景图像（或其它对象），输出在最底层。"子"可以附加到任何 band 中，除页脚外在它们之后输出。你可以附加你的子女到"子"中。大体可以用来打印多选 bands（即 bands 由多个部分组成）。每个部分可以在一个"子"band 中定义并高度可变，高度依赖于对象在一个部分中开始位置。



简单报表（列表）

这是一个比较简单的报表。要创建它，你放置"主项数据"band 到页面上，然后在它上面放置所需的对象：





你可以连接﹃主项数据﹄band 到适当的数据集︵TfrDBDataset 或 TfrUserDataset 控件︶︒这个例子和其它的报表你可以从演示中看到︒

主︑从报表

要创建这个报表︐放置﹃主项数据﹄和﹃细项数据﹄band 到页面上︐然后在它们上面放置适当的对象︒这个报表中使用的数据表必须建立主︑从关系︒

没关系︐你可以随意改变放置的 band 的次序︑﹃主项数据﹄band 将在最开始打印︒如果相应的细项数据列表是空的︐主项数据将移到下一条记录︒如果你不希望这样︐选中﹃主项数据﹄band 的﹃细项数据为空时打印﹄选项︒

你可以在新的一页中打印主项数据︑只要设置 band 的﹃强迫换页﹄选项︒

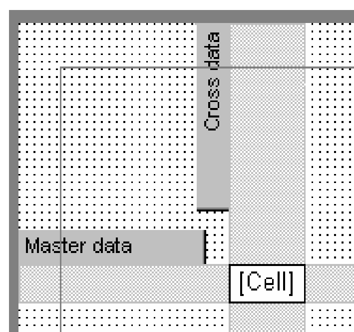
主︑细︑子细报表

要创建这个报表︐放置﹃主项数据﹄﹑﹃细项数据﹄和﹃子细项数据﹄band 到页面上︐然后在它们上面放置适当的对象︒这个报表中使用的数据表必须建立主︑从关系︒

交叉报表

这种报表用于打印栏目可变的表︒在报表生成期间︐所有范围之外的栏目将打印在新的页面上︵象 MS Excel︶︒

要创建这个报表︐放置﹃主项数据﹄和﹃交叉表数据﹄band 到页面上︒放置一些 band 到交叉中︒这个对象将作为交叉表的单元打印︒

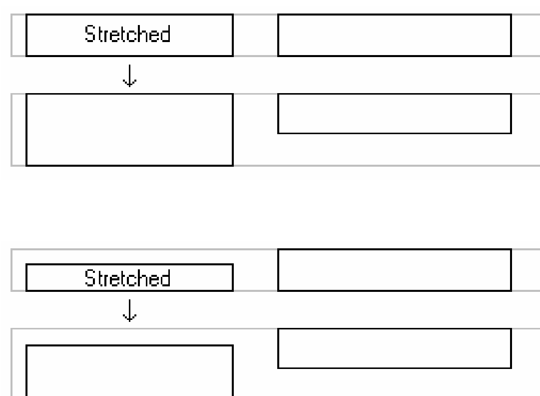


这就是作为基础报表所要做的。你仅需要为“主项数据”和“交叉数据”指定数据源。如果“交叉数据”包含多个普通 band，那就需要为每个 band 指定数据源。对于你来说至少需要两个数据集，并且设置它们之间的主/从关系。

这种报表能够动态改变数据行的高度。你必须在“主项数据”band 中激活“伸展”选项。交叉表将由两次生成：第一次确定最大行高，第二次输出数据。当前的 FastReport 版本不支持构建复杂的交叉表关（例如在 ReportBuilder 中）。这将在 FastReport 的下一个版本提供。

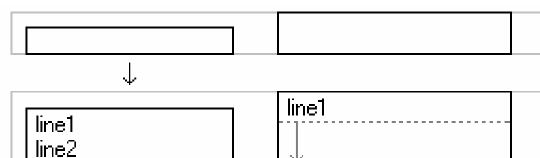
动态报表

动态报表中 band 的高度依赖于 band 中包含的对象。对象的伸展由文本的长度决定，这些文本包括：“Text”，“Text with Shadow”和“Rich Text”。“伸展”选项必须被激才允许这些对象伸展。除了你需要在这些 band 上激活同样的选项外，还确定这些对象已经被放好。当打印一个伸展 band 时，FastReport 了解对象的最大高度，找到 band 并设置它们的高度为对象的最大高度。

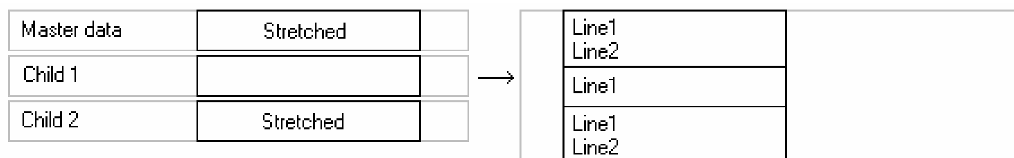


伸展的对象和对象调整高度

如果有多个伸展对象并排，所有对象的底边（与实际高度无关）将与最底的 band 对齐。



你是否考虑过伸展对象放在所有其它对象的下方或非伸展对象放在伸展对象下方呢？在 ReportBuilder 中可以使用使用“Regions”和“StretchWithParent”和“ShiftWithParent”对象属性。对象，放在其它的下方，一个或多个开展伸展，需要转移到不同的 band。

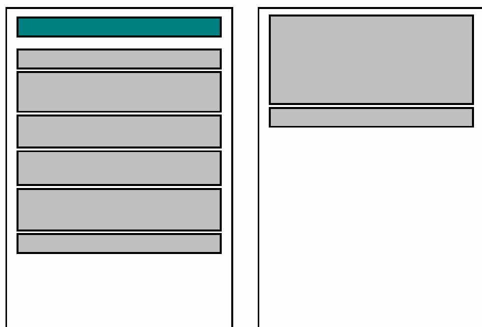


对于所有子 band 你可以设置“伸展”选项。Band 在对象观察器中设置 ChildBand 的属性。在上面的一些例子中你必须设置主数据到 Child1，Child1 设置到 Child2。

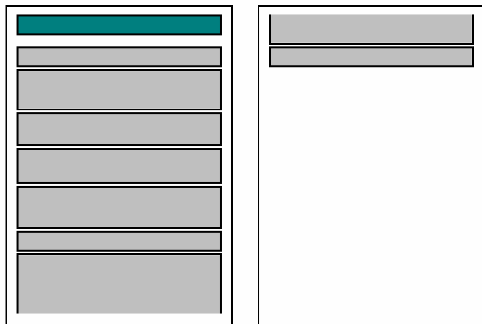
应该让 FastReport 的所有对象有边框。通过设置边框你可以非常方便地得到表格。当一个对象开展伸展时，边框也会伸展。这是非常方便的，因为 ReportBuilder 中的文本对象没有边框，你需要使用“图形”才能得到边框。这有一些不同：你必须对齐边框和对象。此外，你还可以在打印时与附近的边框联接在一起。QuickReport 中也有边框，但不能与对象一起伸展。

bands 分隔

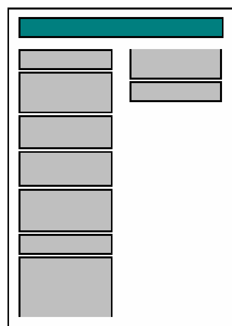
当打印一个 FastReport 的 band 时，如果在页面上有空间，但空间不够，它将生成新的页面并打印在上面。而且页面空间没有用尽，尤其在 band 的高度非常大的情况下。



为了在页面上打印多行文本，band 不必激活“伸展”选项，但必须激活“分隔”。之后报表大概是这个样子：



如果报表是多栏的，对象中的内容将延续到下一栏：



“Text”，“RoundRect”和“RichText”对象能够分隔内容。其它所有对象当没有足够空间时将输出到下一页。

分栏报表

一般的报表生成时都是一页接一页。在分栏报表中将打印在同一页，只不过转到下一栏。在页面选项中设置分栏数。你只需在页面选项中设置分栏数就可把普通报表转为分栏报表。“栏目头”和“栏目脚”将为每个分栏提供一个头和脚。

此外，在 FastReport 中你还可以分别指定各个 band 的分栏数。如果我们为 band 设置 Column>1，数据行的排列将从左到右，从上到下。你也可以设置栏宽和间距。Bands 将显示当前设置的参考线。

带封面的报表

在 FastReport 中创建多页报表（例如，首页为封面，接着是其它信息）。要创建或删除页，选择工具栏和菜单中的功能项，或者在页标上点击鼠标右键调用快捷菜单。

事实上多页报表就是在一个报表中包含多个报表。每个报表又包含自己的页面，band 和页面设置。类似于使用 QR 中的复合报表或 ReportBuilder 中的“SubReport”。

页面内容可以在上页的空白区打印，只需在页面设置中激活“PrintToPrevPage”选项（PrintToPrevPage 在对象观察器中）。

嵌套报表（子报表）

嵌套报表是在报表中放置“子报表”对象。“子报表”对象连接到另一个报表，放在其它页中。报表生成时，由设置的报表替换“子报表”对象。当你插入一个子报表对象时，你将看到在报表中增加了一个新页面。

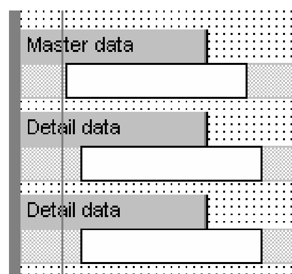
子报表对象可以一个接一个。如果放置子报表到其它在下面，在伸展的 band 中将会伸展它（事实上，你也可以使用子 band 做到这些）。

使用子报表时有一些限制：

- 不能在子报表中使用分栏；
- 不能使用报表标题、报表合计、页头、页脚、栏目 XXX Bands；
- 不能分隔 bands
- 不能使用分组。

主-从-从报表

要创建这种报表，在页面上放“主项数据”和两个“细项数据”bands。联接所有的 band 到相关的数据集上。



你也可以创建“主-从-从-从”、“主-主”、“主-从-子从-子从”报表等等，除了“主-从-主-从”报表。这样的报表必须是多页面-你必须把主-从结构分别放在不同的页中。

复合报表

复合报表是包含多个报表的报表。要创建复合报表，你使用 TfrCompositeReport 对象。将它放置在窗体上并在运行时设置“Reports”属性以包含其它报表。报表将依次打印。如果一个报表页面设置“接上页打印”选项，它打印到上一个报表最后一页的剩余区域。

包含 BLOB 字段报表

如果你想在一个对象中显示 BLOB 数据，只要插入一个相关字段到对象的 memo（或在对象观察器中设置“DataField”属性）。如果你想从非数据库的数据源中显示 BLOB 数据，你可以在 OnBeforePrint 事件中做到这点。

没有 band 的报表

如果希望打印自由表格报表，只包含一个记录的数据，你可以不用 band。所有对象直接放在页面上，并按摆放位置打印。

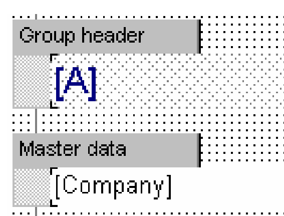
分组报表

分组用于以分组数据为标准进行分组。你可以使用 FastReport 表达式作为分组条件（通常使用

基于 DB 字段的表达式）。当这个表达式的值变化时，FastReport 窗体开始新组。

要创建这种报表，放置“分组头”和“主项数据”band 到页面上。指定合适的数据集到“主项数据”band。在“分组头”的编辑器中输入表达式。例如，按客户名称的第一个字母对客户列表进行分组，输入这个表达式：Copy([CustomerName], 1, 1)，[CustomerName]是指定的数据字段。

报表窗体　　　　　　生成的报表



V
Vashon Ventures 743
VIP Divers Club 32 M
W
Waterspout SCUBA Center 7865

使用分组有一些限制：不能在子报表中使用分组。

注意：用于“主项数据”的数据集必须已按分组条件排序。你可以在查询中加入 ORDER BY 声明来做到这点。

图表报表

假设我们有下表：

Action Club	$1000
Action Diver Supply	$12000
Adventure Undersea	$5000

要创建显示图表的报表至少需要一个象主项数据的数据 band 并指定到适当的 TfrDataset。Band 有不同的矩形对象放在上面。在我们这个例子中，将有“Name”和“Amount”两个字段放地上面。这些对象，事实上，是在 Memo1 和 Memo2 中调用。在 band 上放置 Chart 对象，在我们的例子中将在所有主项数据之后显示，它可以是报表合计 band。

在图表对象的编辑器中选择图表类型并在显示页中配置选项，在数据页中输入用于图例的 memo 的名称（这个例子中是 Memo1），然后输入你要生成图表的数据的 memo 名称（memo2）。

如果数据比较多，你可以选择“前十个”分组值到一个小一点的数字。这将在图表中只显示前几个数据。调整图表连框的大小将改变图表的显示尺寸。预览报表就可以看到结果。这儿有一个说明，要显示图表你在方案中使用变量的数据类型必须是已经支持的。

设计器

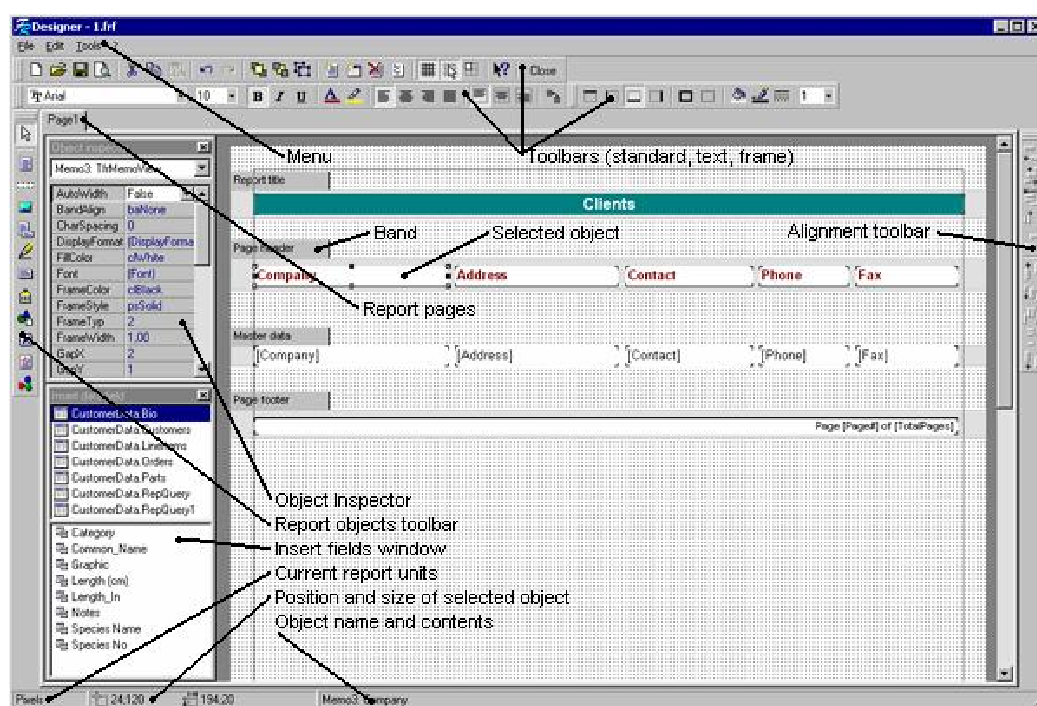
设计器

设计器

FastReport 使用自己的报表设计器，在设计时只要在 TfrReport 控件上双击就可以使用它。设计者允许快速而简单地使用报表设计器并能够在 Delphi 的 IDE 中直接预览报表。

设计器包括可融合的面板（工具栏），可以移动并改变为你所需要的样子。当设计器关闭时位置和可视信息保存在 Windows 的注册表中，并能够在下次打开设计器时恢复。

如果你想在运行时使用设计器，你需要在项目中的窗体上包含 TfrDesigner 控件（用于显示 TfrReport 控件内容）。或者，你也可以在你在单元中引用 FR_Desgn 单元。TfrDesigner 控件能够让你的最终用户调用、保存、设计和修改报表。



使用控制键

- 光标键 — 移动到下一个对象；
- Ctrl + 光标键 — 沿光标方向移动选中的对象；
- Shift + 光标键 — 沿光标方向增加或者减少所选对象的尺寸；
- Enter — 调出所选对象的编辑器；
- Del — 删除所选对象；
- Insert — 显示“插入数据字段”工具栏；
- Ctrl + Enter — 调出所选对象的 memo 编辑器；
- Ctrl + 1..9 — 主，设置所选对象的边框宽度；

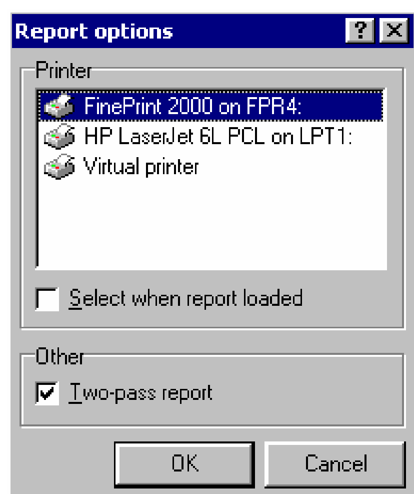
- Ctrl + Z ： 取消最后一次操作；
- Ctrl + Y ： 重复已取消的操作；
- Ctrl + G ： 对齐风格开/关；
- Ctrl + B, Ctrl + I, Ctrl + U ： 设置加粗、倾斜、下划线字体风格；
- Ctrl + F ： 取消对象的边框；
- Ctrl + D ： 设置对象的边框；
- Ctrl + X ： 剪切到剪贴板；
- Ctrl + V ： 从剪贴板粘贴；
- Ctrl + C ： 拷贝到剪贴板；
- Ctrl + A ： 选中页面上的所有对象；
- Ctrl + N ： 创建新的空白报表；
- Ctrl + O ： 打开报表文件；
- Ctrl + S ： 保存报表文件；
- Ctrl + P ： 预览报表；

使用鼠标

- 左击 ： 在页面窗体中选择对象；在可见控件面板选中对象后按左键在页面窗体中插入新对象。
- 右击 ： 调出所选对象的快捷菜单。
- 双击 ： 调出所选对象的缺省编辑器。在页面的空白区域双击则调出页面选项对话框，这里你可以设置页面选项，象页边距，尺寸。
- Shift + 左击 ： 选择多个对象。
- Ctrl + 左击 ： 画选择框。在你松开鼠标后选择框中的所有对象将被选中
- 要按比例缩放所选对象，拖动所选对象组右下角的红色矩形块

报表选项

要设置报表选项，从设计器菜单中选择"文件|报表选项..."。



上面的对话框列出了所有系统中可用的打印机 如果你的系统没有安装打印机 你可以选择﹃虚拟打印机﹄选项 ’这将可以使用所有的打印尺寸 ’但不能打印输出 °这只有在报表的设计状态中可以使用 ’并允许预览报表 °你也可以使用虚拟打印为某个打印机设计报表 ’这不需要在你的电脑中按着打印机︵例如 ’你想设计 A3 幅面的报表 ’但你的报表只能打印 A4︶°

如果选中﹃装入报表时选择﹄选项 ’打印信息保存到报表中 ’当这个报表再次调入时 ’保存的打印机将自动被选择 °如果这个打印机在系统中没有找到 ’那么将选择缺省的 Windows 打印机 °

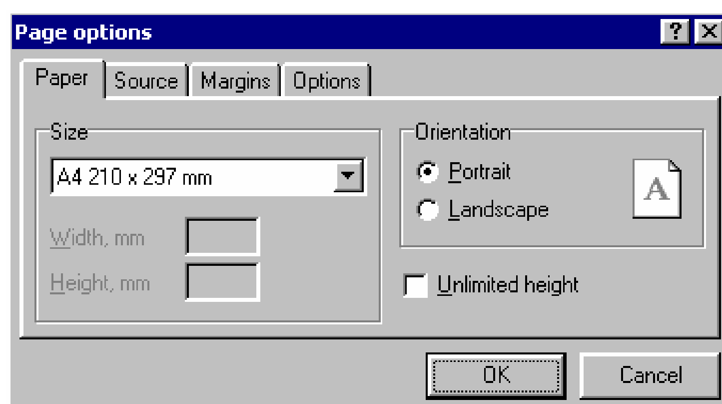
如果你需要在报表中使用﹃总页数﹄函数 ’象打印﹃第 xx 页 共 xx 页﹄时需要选中﹃两遍报表﹄选项 °如果你使用 TotalPages 函数 ’但你忘记选取中这个选项 ’你将在你的报表中得到总页数为 0 °还有一些好处是选中﹃两遍报表﹄可以在最后一次中显示第一次的计算结果 °其中之一就是在分组头中显示分组数 ’可以在演示报表中找到例子 °

在选择打印机之后 ’报表设计器中的页面窗口显示纸张的可用的打印区域和选中的打印机 °

页面选项

要为报表的当前页设置页面选项 ’在设计器菜单中选择﹃文件—页面选项…﹄’或者双击页面的空白区域 °对话框有四页在 °

纸张

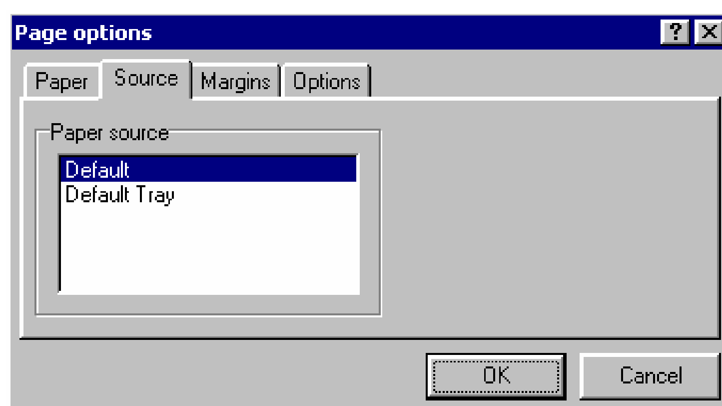


你可以从纸张格式的下拉列表框中选择当前打印杨所支持的纸张格式 °如果当前打印支持自定义纸张格式 ’你就可以从列表中选择﹃自定义﹄’你需要输入自定义纸张格式的宽度和高度 °其它

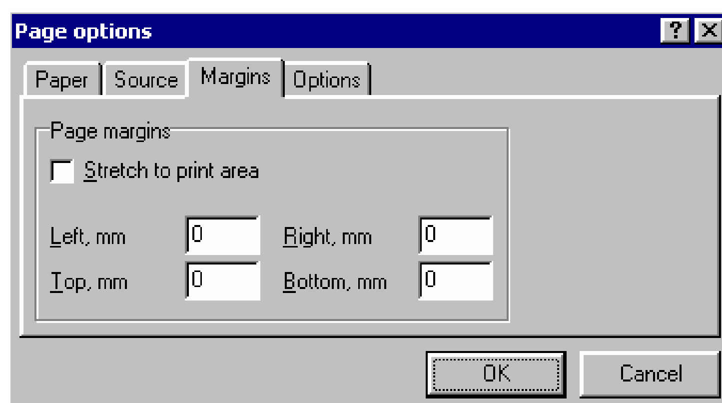
的选择纸张格式，你能够确定纸张方向和设置“Unlimited height”（无限高度），增加纸张高度（如果你想在页面上放置更多的 band）。

注意：不是所有的打印机驱动或打印机都支持自定义纸张格式。（例如：打印机驱动“HP LaserJet 6L”就不支持定义为 76*127mm 的尺寸；打印机驱动“HP LaserJet 4L”不支持所有的自定义尺寸）。

进纸源



边界

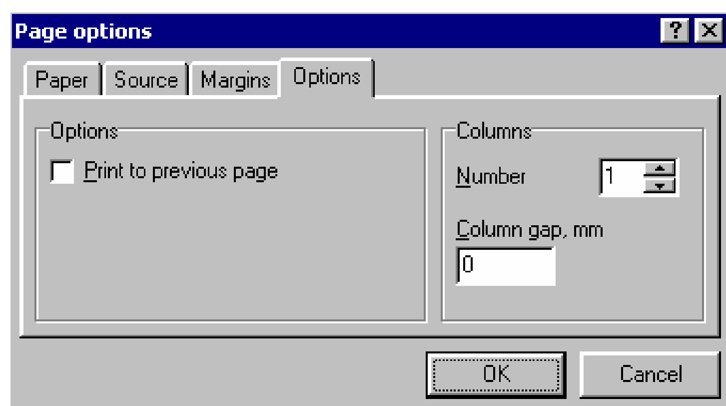


在对话框的“边界”页中，你可以决定是否定义边界和边界尺寸。如果“伸展至打印机”选项被选中，在设计器中页面窗体就不会显示边界区，页面所有区域都会被正确打印。但对象尺寸在不同的打印上是不同的。

如果你取消这个选项，并且所有边界设置为 0，则页边距将自动设置为所选打印机的最大打印区域。当你将设计好的报表从一台打印机转向其它打印机时就会发现这非常有用，特别是喷墨打印机比针式打印机的可打印区域小一点。

如果边界设置为非 0 值，页边距将直接从设计器的页面窗体中反映出来（用灰色的线标出）。如果你使用针式打印机，请注意是否在打印区域中：一些针式打印机将不打印超出打印范围之外的部分，其它的将提示打印超出范围。这显然表明你必须检查一下你的报表。在这种情况下，你只能手工设置边界。

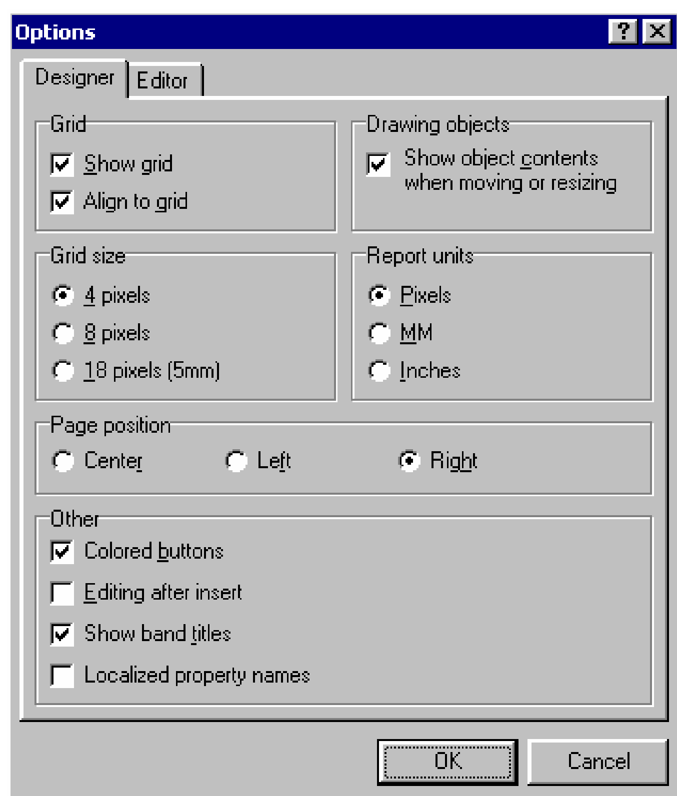
选项

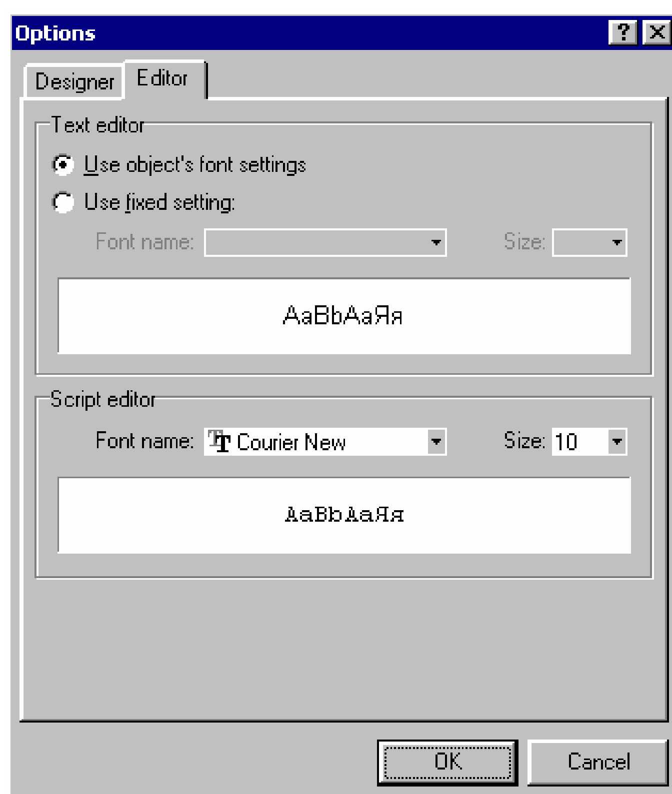


在对话框的"选项"页中，你可以设置页面的有关选项。你可以按照页宽设置分栏数和栏间隔。如果"打印到前一页"选项选中，它允许新的一页从上页的剩余区域开始打印。

设计器选项

从报表器的菜单中选择"工具—选项…"设置缺省选项。





这儿你可以设计栅格大小 ’报表度量单位 ∶象素 `毫米和英寸 °栅格大小 18 象素等于 5 毫米 °

你也可以控制在移动对象时如何显示对象 ∶显示边框或显示内容 °

﹁页位置﹂组允许你选择页面位置︵这对于对象观察器来说是需要的︶°

如果﹁彩色按钮﹂关闭 ’则所有按钮都为黑白色 °

﹁插入后编辑﹂选项设置在插入对象后的动作 ’在对象插入后是否显示缺省编辑器 °如果想插入大量的矩形框时这个选项需要关闭 °

﹁显示 band 标题﹂允许你决定在设计时在页面上是否显示 band︵分页︶的标题 °

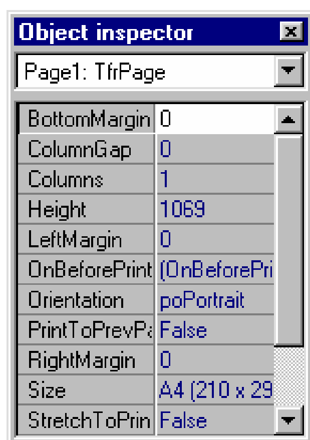
﹁本地化属性名﹂允许本地化对象观察器 °

﹁对齐栅格﹂在移动对象时对齐它们到最近的栅格点 °

在﹁编辑器﹂页中你可以选择编辑器窗口的字体 °你可以在固定设置或对象字体设置之间进行选择 °

对象观察器

对象观察器允许你熟练地设置对象属性 °

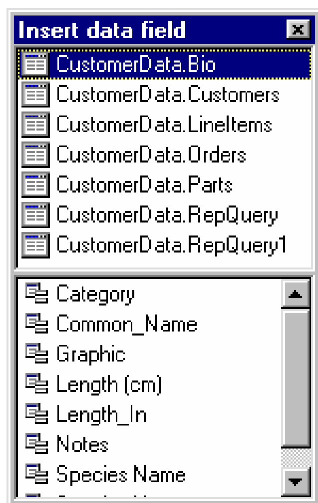


观察器的工作方式与 Delphi 的对象观察器类似。象其它的工具栏一样，你可以让它显示或者隐藏。要显示对象观察器，选择“工具—工具栏—对象观察器”菜单项。要收缩对象观察器，双击标题栏，再次双击标题栏将展开对象观察器。

“插入数据字段”窗口

你使用这个对话框可以快速插入数据库字段到报表中。你可以从菜单“工具—工具箱—插入数据字段”或选择“插入数据库字段”按钮运行它。这个窗口包含两个列表：表（查询）列表在窗口上部，字段列表在下面。

可以采用拖放方法插入字段。你用鼠标从底部列表选择所需源字段并按住不放，然后移动它到页面上。当你松开鼠标时，将出现一个新的“Text”对象并连接到数据库字段。



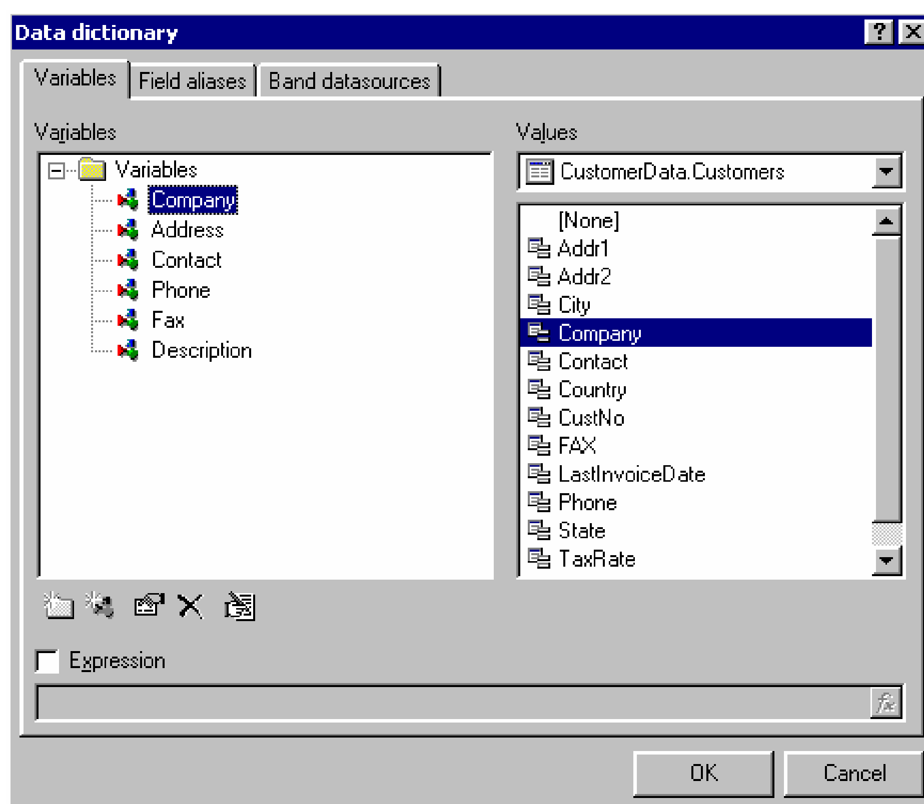
如果对话框经常使用，你可以把它放在对象观察器的上方或下方。这时双击对话框标题栏，窗口将最小化而观察器将最大化；再次双击则相反，它将恢复到正常大小而观察器窗口将最小化。

数据字典

“数据字典”窗口可以从“文件—数据字典…”菜单项中调用。这个窗口看上去象一个多页文本编辑器：“变量”、“数据别名”和“band 数据源”。数据字典保存在报表窗口中，但也可以保存在单独的.FRD 文件中，在需要时再从中读出数据到报表中。要做到这点，在设计器菜单中选择“文

件—打开”和“文件—另存为”并在出现的标准打开或保存对话框中选择文件类型为“FastReport 数据字典文件”。

变量

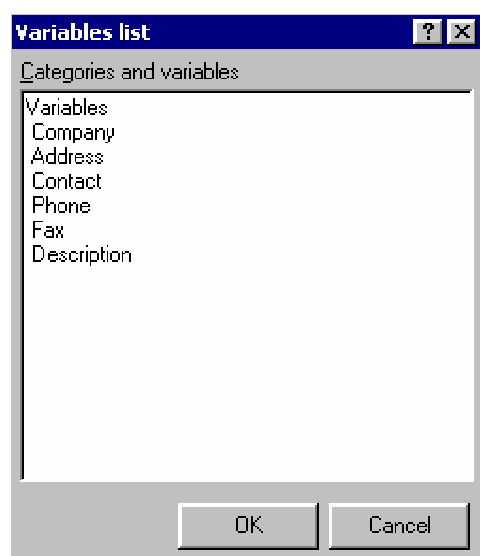


这个页用于操作变量列表。

变量列表在窗口的左边。从图中我们可以看出，列表结构是两层的。由分类组成，在每个分类中必须有一个或多个变量地。分类的按插入顺序排好，但它们不能插入到报表中。

当创建新的空白报表时列表是空的，你可以使用下列按钮来设置：

— 增加新的分类而 — 在当前分类中增加的变量。这两个动作也可以使用快捷键：“Insert”键增加新变量，“Ctrl+Insert”增加新的分类。对于其它图标： — 编辑变量或分类名称（快捷键 —“Enter”）， — 删除一个变量或分类（“Delete”键）。 — 调用变量列表编辑器，它将显示一个字符列表。这里你可以从剪贴板插入多个变量，把变量从一个分类移到另一个分类。



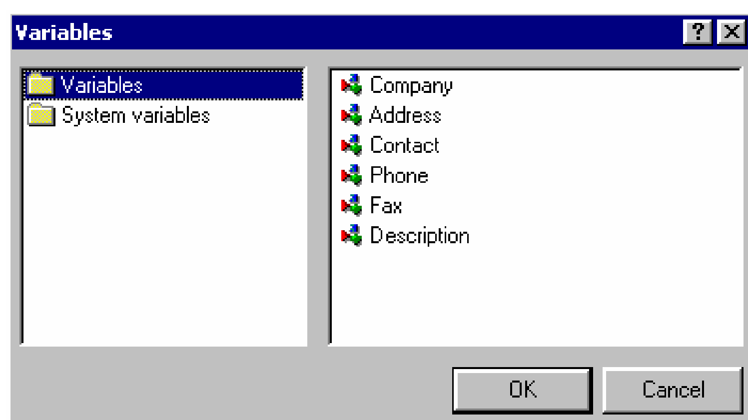
在定义变量列表之后，你可以为各个变量设置值。要做到这一点，你需要用鼠标在左边列表中选择一个变量，在右边列表中选择一人值。

所有的数据源都显示在右边的列表中（非数据感知控件，Tdataset 的子类），在这个时候是可用的，字段的名称同样也是变量。在设计时，所有打开的模块（单元）中的所有数据集都可用。在运行时，所有已创建的窗体或数据模块（TDataModule）中的所有数据集都可用。你还可以从右边的列表中选择“系统变量”，它们可以是下列变量值中的一个：

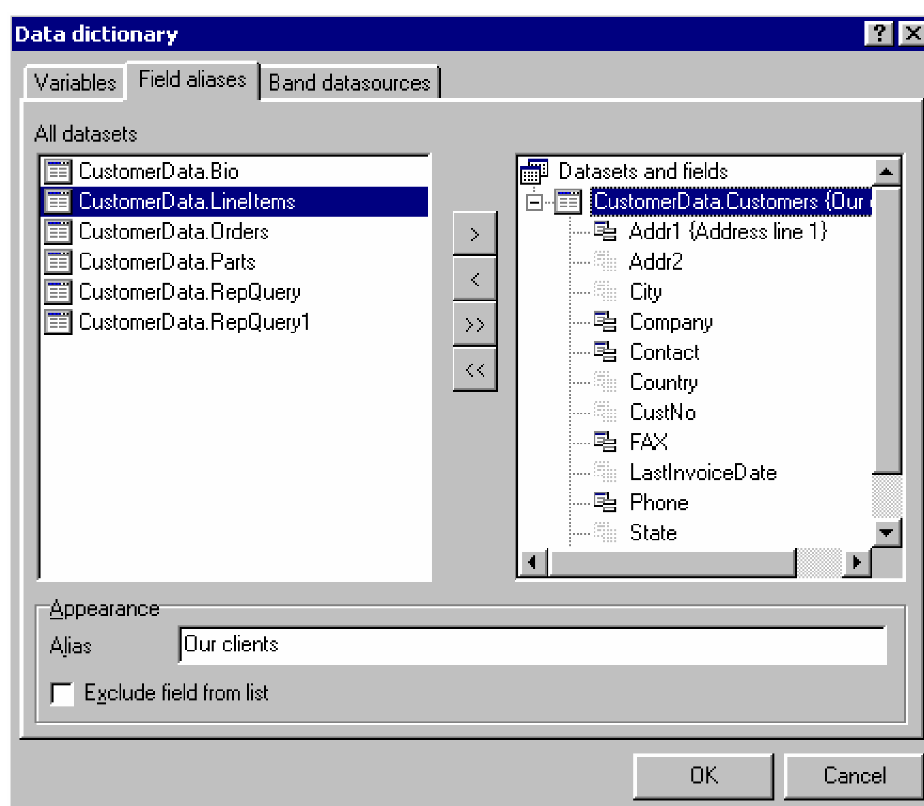
- 页：当前页码，等同于 Page# 函数（参考内部函数描述）；
- 日期：开始创建报表的日期，等同于 Date；
- 时间：开始创建报表的时间，等同于 Time；
- 行，行 Through#，列，当前行#，总页数：参考内部函数描述；

变量的值也可以是一个表达式：选择窗口下方的“表达式”选择框和“表达式”按钮。激活文本字段表达式。要显示表达式按这个按钮 。

变量插入窗口的一个典型例子就象这样：



字段别名



在这个页中’有一个项目中所有窗体中的所有可用数据集的列表ı数据表`查询°

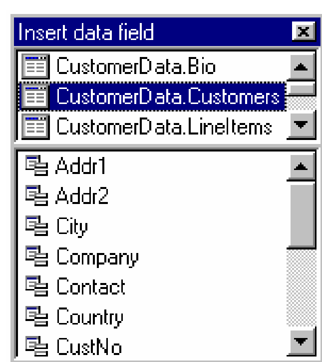
这个对话框用于从列表中去除不需要的数据’并换一个更有意义的名称︵别名︶°这是报表需要的’作为规则’用于区分查询或数据表°FastReport 缺省是直接从所有的数据源中插入字段’从所有的项目窗体中°而在一个大的项目中至少有十个︵或十个以上︶的表或查询°

要使用一个数据集’你只需把它从左边移到右边°你可以使用拖放的方法’或者双击源数据集’或者使用窗口中间的按钮°

要从列表中移去字段’选择源数据集并选中字段’然后选择﹃从列表中排除字段﹄°再次点击这个框将重新包含这个字段°这个操作也可用﹃空格﹄键完成°

要设置别名’选择一个数据源或字段并在﹃别名﹄中输入新的名称°如果别名不需要了ı清除字段的所有数据°这个操作也可以用﹃Enter﹄来完成°

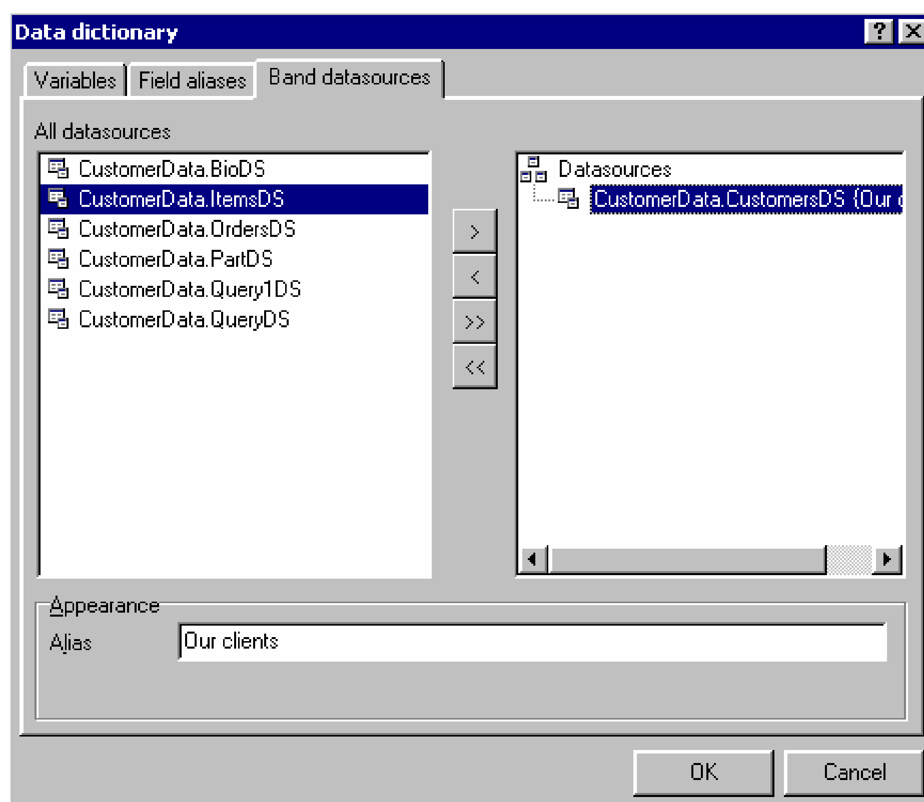
从这个图象中你也可以看出’当构建报表时为数据源和字段创建别名是非常简单的°比较一下插入字段对话框在设置别名和设置别名之后：





没有设置别名　　　　　　　　使用别名

Band 数据源

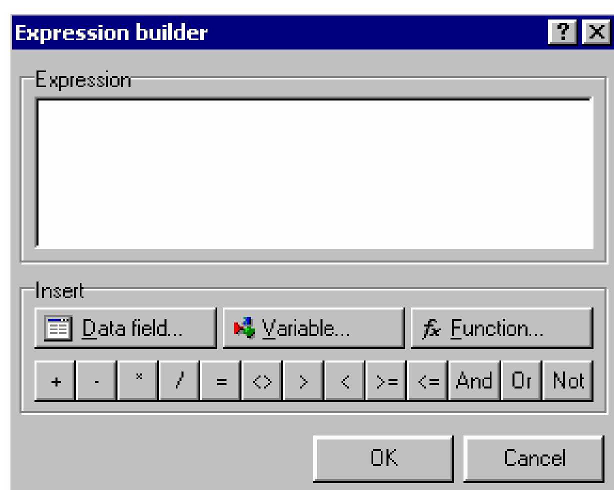


这个页显示对于 band 可用的所有数据源列表，控件 TfrDBDataSet，TfrUserDataSet，能够在项目的所有窗体中能够找到的。象前面的对话框，你可以使用更明确的名称（别名）。结果将显示在为数据 Band 选择数据源的对话框中：



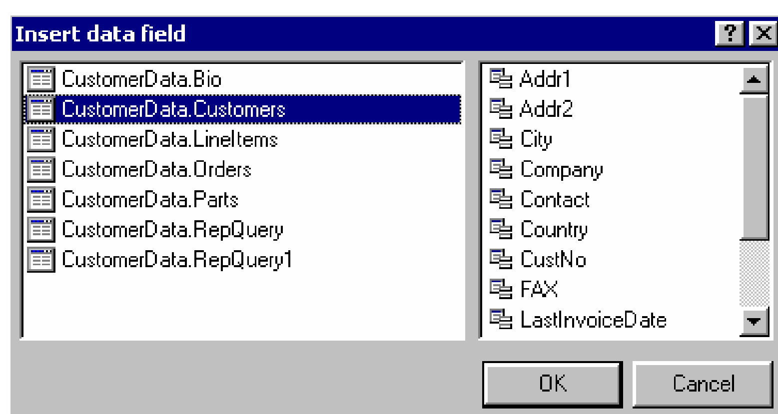
表达式编辑器

表达式编辑器可以从文本编辑器中运行，通过从工具栏中选择。你可以大多数对话框中使用它，如果需要设置一个表达式的话（例如，在分组条件编辑器中，分组头）。



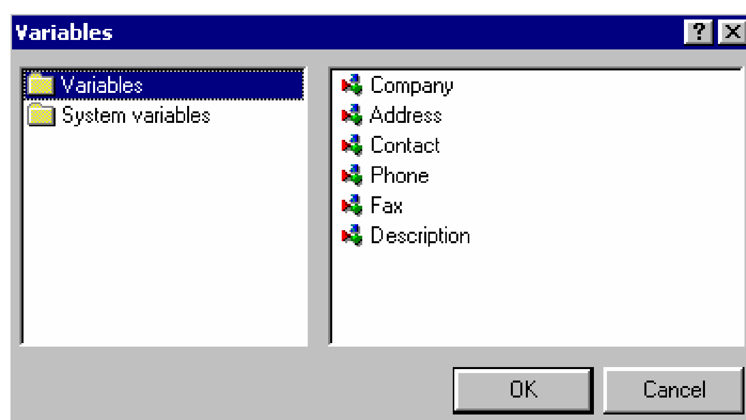
这个窗口包含一个用于输入公式的编辑框，以及一些按钮，用于调用变量、数据字段和函数插入对话框，还有一些用于快速插入算术和逻辑运行符的按钮。

“插入数据字段”对话框



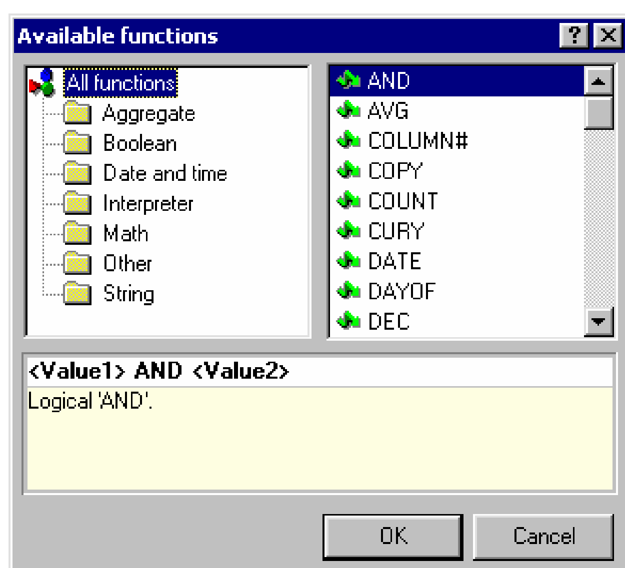
这个对话框用于帮助选择要插入到表达式中的数据字段。左边是所有可用的数据表列表。右边是所选数据表的所有字段。要插入一个字段到表达式中，选择一个字段并按“确认”按钮，或双击字段。

“插入变量”对话框

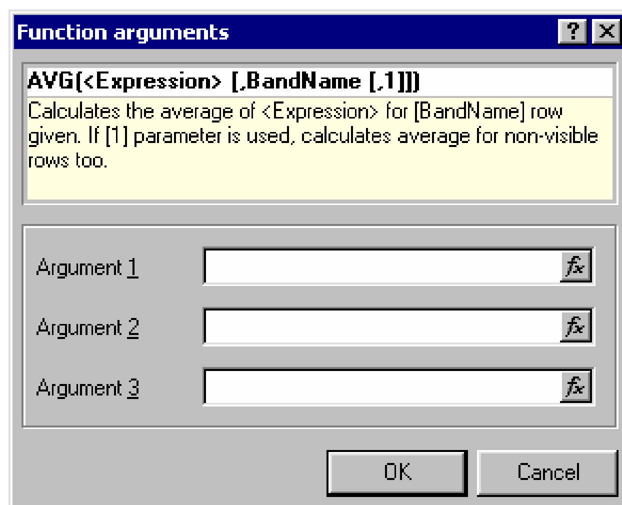


这个对话框用于帮助选择要插入表达式中的变量。左边是分类列表。右边是所选分类的变量列表。要插入一个变量到表达式中，选择一个变量并按“确认”按钮，或双击它。

“插入函数”对话框

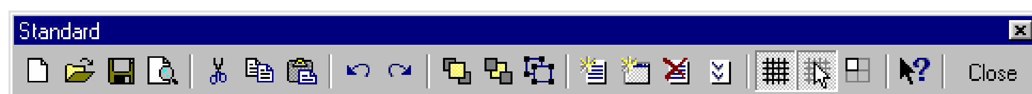


你可以从这个对话框中选择一个函数插入到表达式中。当选中一个函数时，你可以在窗口下方看到函数的一个简短说明。如果函数是所需的，你在回答提问后按“确认”：



工具栏

“标准”工具栏



图标	名称	描　　述
	新建报表	创建新的空白报表
	打开报表	打开 FRF 报表文件。快捷键：Ctrl+O。

	保存报表	保存报表到 FRF 文件。快捷键：Ctrl+S。
	预览	运行报表并在预览窗口显示。快捷键：Ctrl+P。
	剪切	剪切所选的对象到剪切剪。快捷键：Ctrl+X。
	拷贝	拷贝所选的对象到剪切剪。快捷键：Ctrl+C。
	粘贴	从剪贴板中粘贴对象。快捷键：Ctrl+V。
	撤消	撤消最后的操作。最多可撤消 100 次。快捷键：Ctrl+Z。
	重复	重复已撤消的最后一次操作。快捷键：Ctrl+Y。
	移到最前	调整对象到最前。
	移到最后	调整对象到最后。
	全选	选择当前页面上的所有对象。快捷键：Ctrl+A。
	新页	创建的空白页面。
	删除页	删除当前页面。
	页面选项	显示页面选项对话框。
	显示栅格	在页面上显示栅格。栅格的大小可以从设计器的甚而对话框中调整。快捷键：Ctrl+G。
	对齐栅格	当移动对象或重定义对象大小时，位置和大小将按栅格进行改变。
	匹配栅格	当改变所选对象的位置和大小时使它们与栅格匹配。
	帮助	显示选对象的帮助主题。
Clos	关闭	关闭设计器窗口。

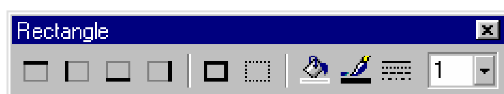
“格式”工具栏



图标	名称	描　　述
Arial	字体格式	下拉列表中包括了系统中已安装的所有字体。双击这个控件，你将可以看到标准的“字体”对话框。
10	字体大小	下拉列表中包括了所选字体的可用字体大小。如果你手工设置大小，点击这个控件，输入所需大小并按回车。
B	粗体	设置“粗体”字体修饰。快捷键：Ctrl+B。
I	倾斜	设置“倾斜”字体修饰。快捷键：Ctrl+I。
U	下划线	设置“下划线”字体修饰。快捷键：Ctrl+U。
A	字体颜色	从下拉颜色面板中选择字体颜色。
	条件颜色	设置按条件改变的颜色

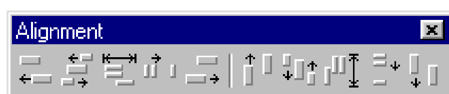
	左对齐	文本按对象左对齐。
	居中	文本按对象的宽度居中。
	右对齐	文本按对象右对齐。
	宽度对齐	文本左右两端对齐。
	顶端对齐	对齐文本到对象顶端。
	垂直居中对齐	对齐文本到对象高度的中间。
	底端对齐	对齐文本到对象底端。
	文字方向	改变文字方向（0 或 90 度）。

「边框」工具栏



图标	名称	描　　述
	顶框线	开/关顶部边框线。
	左框线	开/关左边框线。
	底框线	开/关底部边框线。
	右框线	开/关右边框线。
	全部框线	显示全部边框线。
	无框线	隐蔽全部边框线。
	填充颜色	从下拉面板中选择对象的填充颜色。
	边框线颜色	从下拉面板中选择对象的边框线颜色。
	边框线型	从下拉列表中选择对象的边框线类型。
1	线宽	从下拉列表中选择对象的边框线宽度。

「对齐」工具栏



图标	描　　述
	所选对象按第一个对象的左边对齐（不适用于单个控件）
	移动所选对象按第一个对象的水平中间对齐（不选用于单个控件）
	按窗口的水平方向对齐所选对象。

	水平方向平均分布所选对象，使之间隔相等。
	所选对象按第一个对象的右边对齐（不适用于单个控件）
	所选对象按第一个对象的顶部对齐（不适用于单个控件）
	移动所选对象按第一个对象的垂直中间对齐（不选用于单个控件）
	按窗口的垂直方向对齐所选对象。
	垂直方向平均分布所选对象，使之间隔相等。
	所选对象按第一个对象的底部对齐（不适用于单个控件）

程序设计

事件

数据经常要从非数据库的数据源中提取（例如：文件，数组，等等）。TfrUserDataset 控件可以达到这个目的。它包括下列事件：OnFirst, OnNext, OnCheckEOF。另外 TfeReport 的 OnGetValue 和 OnBeforePrint 也常常用到。

OnGetValue 事件在对象文本中的变量需要数据时触发。可以在这儿处理为内部控件的变量赋值。另外需要在外部加上事件处理，例如：

```
procedure TForm1.Doc1GetValue(const ParName: string; var ParValue: Variant);
begin
  if ParName = 'Var1' then
    ParValue := '1'
  else if ParName = 'Var2' then
    ParValue := 2
end;
```

OnBeforePrint 在画出所有对象之前触发 经常用于调入备注字段的内容或从数据库调入图象到对象中。下面有这样一个例子：

```
procedure TForm1.Doc1BeforePrint(Memo: TStringList; View: TView);
begin
  if Memo.Count > 0 then
    if Memo[0] = '[Memo]' then
      Memo.Assign(Table1Memo)
    else if (Memo[0] = '[Picture]') and (View is TPictureView) then
     (View as TPictureView).Picture.Assign(Table1Picture);
end;
```

OnUserFunction 事件在当对象文件中的变量或表达式包含函数时触发。函数一般有三个任意类型的变量。例如：

```
procedure TForm1.Doc1UserFunction(const name: string; p1, p2, p3: Variant;
  var val: Variant);
var
  d: Double;
begin
  if name <> 'MONEYTOSTR' then Exit;
  d := frParser.Calc(p1);
  val := MyMoneyToStr(d);
end;
```

FastReport 控件面板中的 TfrUserDataset 用于遍历非数据库的数据源（例如：遍历数组）。它 OnFirst, OnNext,包括 OnCheckEOF：它们的用法很明显。

TfrReport 对象的其它事件

- FastReport 除了上面所介绍的一些事件外，还有其它一些事件：
- OnBeginBand：事件在数据部分被定义时触发。这里可以决定哪些数据被生成。

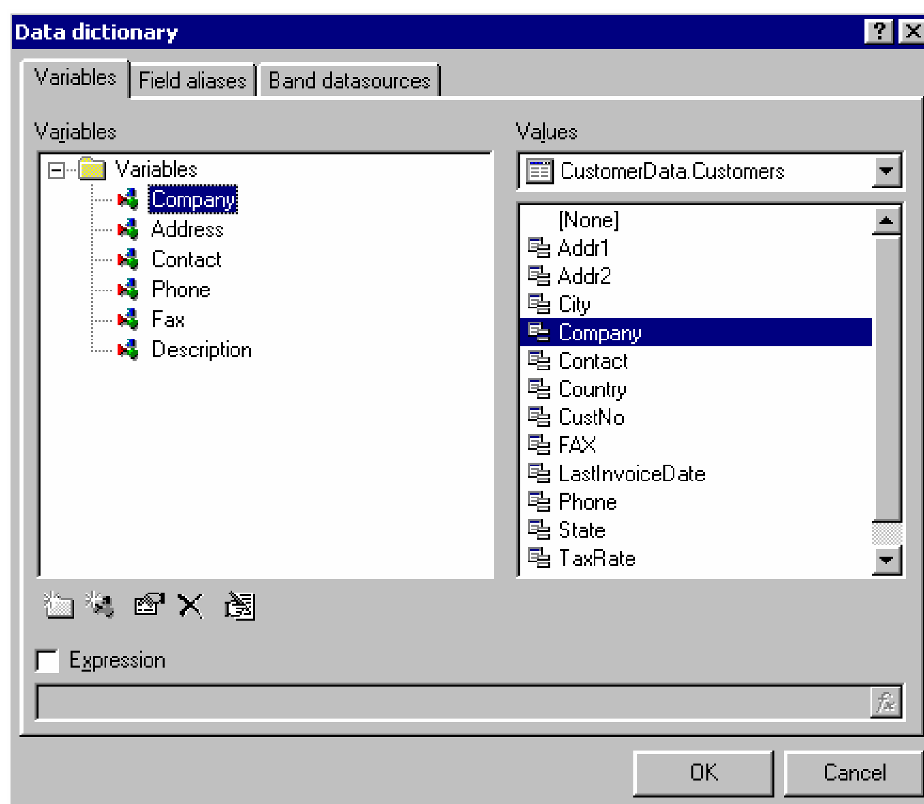
- OnBeginColumn：事件在报表切换不同的分栏时触发。多栏部分的 Band 参数可以在这时处理。这个可用一般用于切换数据源（多栏部分有一个标题，主数据，细数据，等等，可以使用不同的数据源）。
- OnBeginDoc：报表生成之前触发。
- OnBeginPage：页面生成之前触发。
- OnEndBand：在数据部分正确生成之后触发。
- OnEndDoc：在报表生成之后触发。
- OnEndPage：在页面生成之后触发。
- OnManualBuild：在生成报表定制时触发（通过代码）如果事件处理，报表也一样生成。如果报表定制生成，你可以改变输出顺序。
- OnPrintColumn：当生成交叉报表栏时触发。栏目的序号和缺省宽度可以通过这个事件来设置。你可以交叉表的不同栏改变不同的宽度。这时你只要改变报表单元中对象的宽度。
- OnProgress：这个事件在标明当前生成进度时触发（报表创建，打印，输出）。在这个事件你可以使用自己的进度表。

变量

你可以在 FastReport 的对象和表达式中使用变量。这是什么意思？它们有什么用？

假设有一个职员表，包括字段 N，Nname1，Name2 和 Name3。字段的名称在不同的时候作用是不一样的，一般用于数据库设计人员。也就是说，需要重新指定一个更有意义的名称用于插入到对象中。当然，变量不仅能用于数据库字段名称，也能用于算术表达式。一个例子就是对数据字段、数值、日期或时间求和。更多的使用变量的例子可以从报表演示的程序代码中找到。

从操作一个变量列表，你必须从“文件—数据字典...”菜单中打开一个对话框。



变量列表在屏幕的左边。从这个图像可以看出，这个列表是一个两层结构：它由分类和各分类所包含的一个或多个变量组成。分类仅用于对变量逻辑上进行分组，是不能插入到报表中的。有关这个技术的详细说明参考"设计器"部分。

但是，FastReport 数据字典中的变量描述不能单独工作。如果没有定义相关代码，FastReport 会按这样的假设查找：

- 在数据字典中有这样一个变量；
- 它是一个数据字段；
- 它是一个类似于 Page#，Date，Time 或其它类型的专用变量；
- 它是 frVariables 列表中的变量；
- 它是一个对象的属性；
- 它是一个 frConsts 中的常量；

同样，如果使用控件的 OnGetValue 事件，它在所有验证之前调用。如果事件处理返回一个值，变量将被初始化，并且不对它做进一步的验证。

这种方法使得报表中的变量赋值方法具有多样性。它推荐为：

- 当生成过程中将一个不可改变的静态记录值赋值给一个记录时，可以使用一个 frVariables，就象这样：

```
frVariables['Reporting Period'] := 'January';

frReport1.ShowReport;
```

- 当生成过程中将一个可改变的记录值赋值给一个记录时，你可以使用数据字典或

TfrReport.OnGetValue 事件处理，就象这样：

```
procedure TForm1.frReport1GetValue(const ParName: String; var ParValue:
Variant);
begin
  if AnsiCompareText(ParName, 'Reported Period') = 0 then
    ParValue := Table1OtchPeriod.Value;
end;
```

- 也可以通过编程为一个变量列表赋值：

```
with frReport1.Dictionary do
begin
  Variables['Number’] := 1;
  Variables['Sum'] := '0.2 * Table1."Summa"';
  Variables['Date'] := '''' + 'January' + '''';
end;
```

（在这个例子中需要另外说明的是要定义一个字符常量）。

FastReport 中指定给数据字典中的变量的字符串，可以作为一个表达式被计算。常常用来使用一个没有在数据字典中定义的变量。

必须记住的是如果你从 frVariables 列表中使用变量或者你通过 OnGetValue 事件遍历它们，它们不能放进数据字典。

扩展 FastReport 函数

FastReport 生成器是开放的，意味着你可以通过自己的函数库、新的可视控件、输出文件格式和向导来扩展它。这部分包括了进行这种扩展的方法。

设计你自己的预览窗口

有时需要用你自己的预览窗口来替换标准的。这个方法能被用于当在输出前运行附加的函数（例如插入打印数据到数据表中）或去除预览窗口中不需要的部分。

FastReport 允许开发者用任何方法开发自己的预览窗口。FastReport 控件面板上中的 TfrPreview 控件就是为了达到这个目的。这个控件的主要功能有：

- First, Prev, Next, Last—在报表中进行导航：首页、上页、下页、末页；
- SaveToFile—保存已生成的报表到任何支持的文件格式中；
- LoadFromFile—从*.FRP 格式的文件中调入已生成的报表；
- Print—打印已生成的报表；
- OnePage—设置预览的比例为一页大小；
- PageWidth—设置预览的页面宽度。

除了上面可用的功能外，TfrPreview 还有一个 Zoom 属性用于设置缩放显示。这个缩放显示可以按原始尺寸的比例进行缩放。TfrPreview 控件还将显示滚动条和用于显示在文档中的当前页和总页数的状态栏。

要替换标准预览窗口到 FastReport 报表生成器，TfrReport 控件的预览属性必须相应地改变为这个窗口。你可以从 FastReport\Examples\目录中的\Report\子目录中找到使用自定义预览窗口的例子。

扩展函数列表

这个手册很早就描述了怎样扩展 FastReport 的函数功能。这个方法是基于 TfrReport 控件的 OnUserFunction 事件。这个方法的缺点是：所有的 TfrReport 对象的自定义函数都必须放在同一个 OnUserFunction 事件处理中（也就是自定义函数都必须到 TfrReport 中去找）。作为选择，你也可以将一系列的函数放到一个单独的类库中（函数类库）。

FastReport 支持这种可选方法，你可以写一个 TfrFunctionLibrary 的继承类（继承什么？）或者可以使用（引用）。这个类中定义了一些开发者需要了解的 FastReport 工作的基本属性和方法。

这是 TfrFunctionLibrary 在 FR_Class 模块中的声明：

```
TfrFunctionLibrary = class(TObject)
public
  List: TStringList;
  constructor Create; virtual;
  destructor Destroy; override;
  function OnFunction(const FName: String; p1, p2, p3: Variant;
    var val: String): Boolean;
  procedure DoFunction(FNo: Integer; p1, p2, p3: Variant; var val: String);
    virtual; abstract;
end;
```

下面是一些关键模块：

Create：设计器生成一个可用的函数列表，并按大写字母顺序；

DoFunction：当调用时，这个方法用于返回（返回什么？）使用参数计算的函数值。每个函数最多有三个参数。这些参数的值是通过 DoFunction 的 P1，P2，P3 参数取得。

下面是一个有两个函数的简单例子：

```
  TMyFunctionLibrary = class(TfrFunctionLibrary)
  public
    constructor Create; override;
    procedure DoFunction(FNo: Integer; p1, p2, p3: Variant;
      var val: Variant); override;
  end;

constructor TMyFunctionLibrary.Create;
begin
  inherited Create;
  with List do
  begin
```

```
    Add('DATEPROPIS');
    Add('SUMPROPIS');
  end;
end;

procedure TMyFunctionLibrary.DoFunction(FNo: Integer; p1, p2, p3: Variant;
  var val: Variant);
begin
  val := 0;
  case FNo of
    0: val := My_DateConvertion_Routine(frParser.Calc(p1));
    1: val := My_SumConvertion_Routine(frParser.Calc(p1));
  end;
end;
```

要在 FastReport 中激活已定义的函数类库，你需要注册它，意思是调用 frRegisterFunctionLibrary 方法并将这个类库作为它的一个参数（不是类的实例）。例如：

```
frRegisterFunctionLibrary(TMyFunctionLibrary);
```

要增加附加函数列表以"插入函数"对话框中，你需要通过下列参数调用 frAddFunctionDesc 方法：

函数名称；

这个函数的分类名；

这个函数的语法和目的的描述。注意：在函数描述中必须有"/"标记，它用于在函数中分隔语法描述。

记住每个函数都需要单独调用 rAddFunctionDesc。这里有一个在表达式生成器中注册函数的例子：

```
frAddFunctionDesc('SUMPROPIS', 'My functions',
  'SUMPROPIS(<Number>)/Returns value spelled out.');
```

报表示例

报表示例

报表示例

FastReport 配套有一组按不同的操作和不同的报表类型顺序的示例。这些示例文件可以从 FastReport 主目录中的 DEMO 子目录中找到。

标准套件中有 9 个不同方面的示例项目：

- 在报表中插入图表（Chart 目录）；
- 存储临时报表到 Delphi 的窗体中，而不是 FRF 文件（DFMSTORE 目录）；
- 组织"最终用户"工作环境（USER 目录）；
- 使用数据感知控件（ENDUSER1 目录）；
- 运用 OnManualBuild 事件控制报表逻辑结构（MANUAL 目录）；
- 运行时使用代码手工合成报表（RUNTIME 目录）；
- 打印可变栏数或未知栏数的报表（PRNTBL1 目录）；
- 打印可变栏数和栏宽可变的报表（PRNTBL2 目录）；
- 使用标准和 Delphi 设计器和开发环境创建不同的报表（REPORTS 目录）。

这里所提到的所有示例可以很好地指导那些想在短时间内找到创建大多数的可能用到的新报表而不想重读整个文档的开发者。

让我们来好好地看看这些例子。

在报表中插入图表

这个例子可以在 FastReport 示例目录中的 CHART 子目录中找到。这个项目仅包含一个窗体，以及窗体中的下列控件：

1．一个 Ttable/TdataSource 控件用于从数据库中提出数据（这个例子使用的是标准 Delphi 示例中的 COUNTRY.DB 作为数据源）；

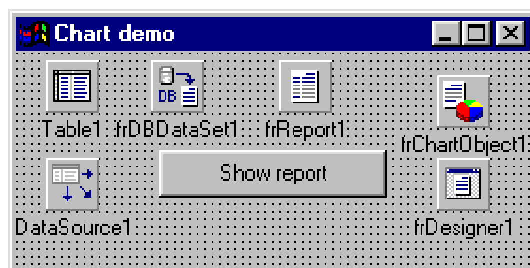
2．TfrDBDataSet 控件用于导入一个 DB 数据源到 FastReport 对象；

3．TfrReport 控件－报表自身；

4．TfrChartObject 控件－"图表"单元；

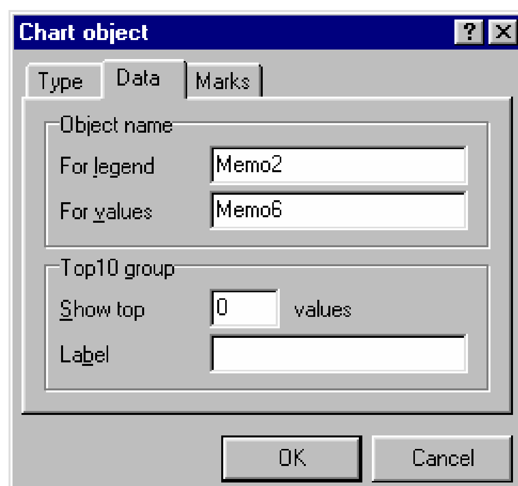
5．TfrDesigner 控件－运行时报表设计器（最终用户报表设计器）。

这个窗体看起来就象这样：



显示的报表包含三页，每页上都有一个图表。第一个在垂直轴上显示 COUNTRY 表中的 AREA

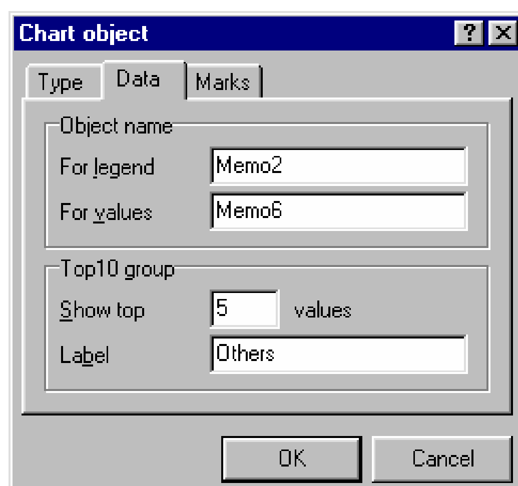
字段内容。第二个显示类似的数据，只不过是使用 POPULATION 字段。为了让两个图显示包含一些数据的“图表”对象，必须放置“图表”对象到报表窗体上，并将数据指定到它。这可以通过在对象双击激活的“图表”对象编辑器来做。在这个编辑器的“数据”页中，“对象名称”组中的“图例”和“值”必须填好。



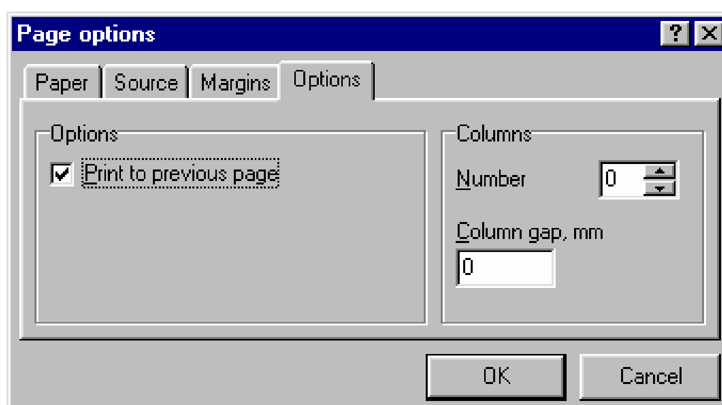
这个例子中的这些 Memo2 和 Memo6 包含这些字段的值。Memo2 和 Memo6 是“Text”对象，用于连接到 COUNTRY 数据表的 AREA 和 POPULATION 字段。

因此，所有 COUNTRY 数据表记录的相应字段的数据将被输入到“图表”对象中。

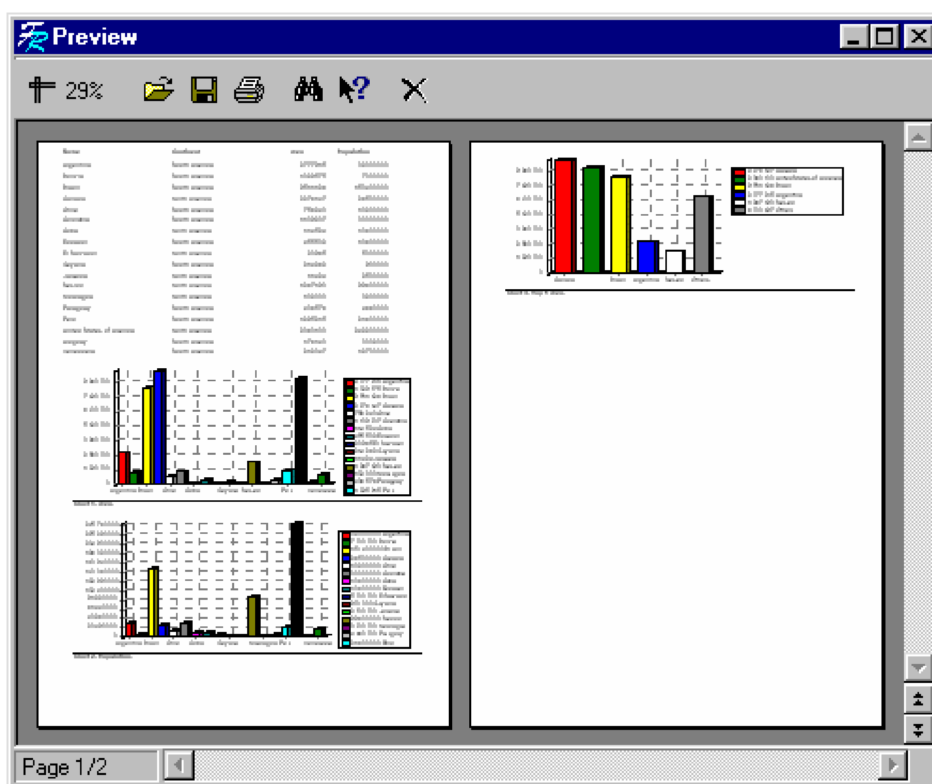
第三个图表示例中使用了“前 X 个分组”，仅显示前 X 个值。其它值的总数放在名称为“其它”的单独的栏中。这需要在对象编辑器的“数据”页中的“项端显示”字段中设置非零值，并在“标号”字段中输入其它值的求和项的名称。这个图中显示设置前 5 个报表的控件编辑器对话框。



这个报表让所有的页面都接着各自的上一页打印。这使生成报表页面更有效。这个特性需要在页面属性对话框中设置适当的标记。



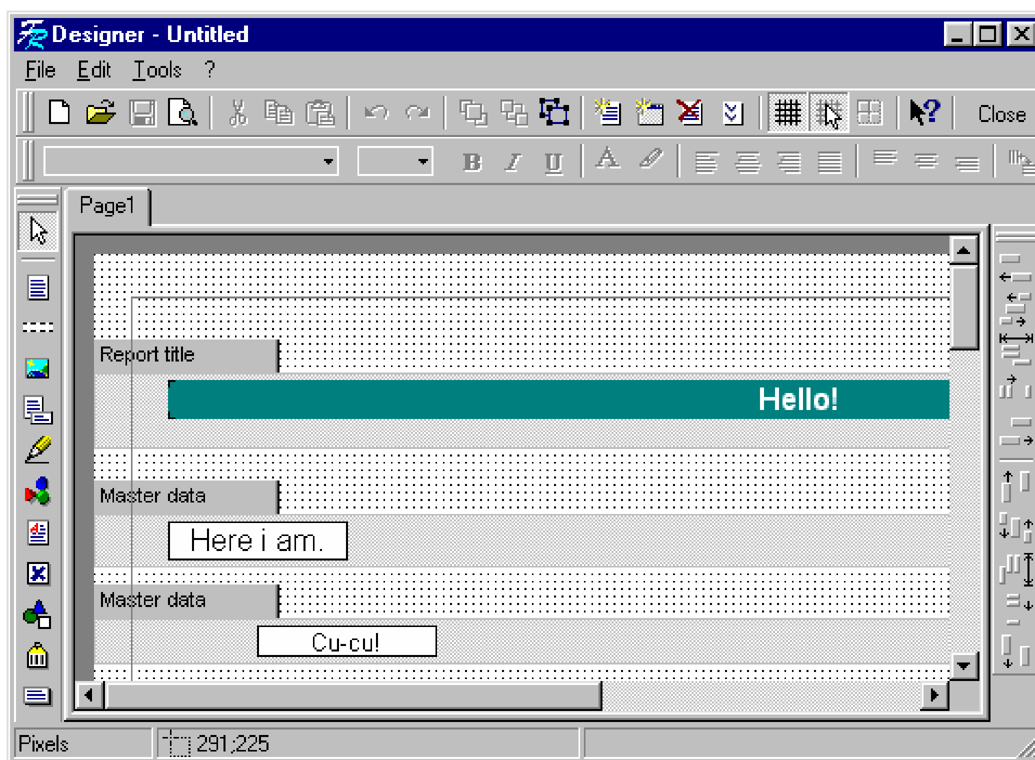
报表的预览窗口显示如下：



运用 OnManualBuild 事件控制报表逻辑结构

这个例子的源代码放在 MANUAL 子目录中。

这个例子只包含一个只有一个用于报表预览按钮的简单窗口。这个报表窗口在 FastReport 设计器中看起来就象这样：



如果你按设计器中的预览按钮，你就会看到"Hello!"行在绿色的背景上。这是因为两个"主项数据"区域没有连接到任何数据源，所有它们是不打印的。

然而，如果你执行这个例子，你将得到四页报表，两个"主项数据"区域被打印，并包含其中的文本行。这需要通过下面的 FastReport 对象的 OnManualBuild 事件处理中才能实现：

```
procedure TForm1.frReport1ManualBuild(Sender: TfrPage);
var
  i, j: Integer;
begin
  Sender.ShowBandByType(btReportTitle);
  for i := 0 to 3 do
  begin
    Sender.ShowBandByName('Band2');
    for j := 0 to 2 do
      Sender.ShowBandByName('Band3');
    if i <> 3 then
      Sender.NewPage;
  end;
end;
```

这个事件处理中标题部分的所有内容将通过 *Sender.ShowBandByType(btReportTitle)*打印在第一页。接下来四个页面将通过一个循环使用 *Sender.NewPage* 方法并将有"Cu-cu!"文本的区域在每页中打印三次。

这表明可以通过 OnManualBuild 事件处理控制报表内容的逻辑结构。这给报表生成带来弹性。

运行时使用代码手工合成报表

有时报表的结构在应用的设计阶段是不确定的或者它需要在运行时改变。这种情况就必须动态创建报表模板。

象这种情况，报表不固定，所以不能通过程序员为软件定义一个相应的独立文件到一个新窗体中，只能使用最终用户报表。

对于 FastReport，它利用你自己的程序代码动态创建报表窗体，就象 VCL 对象一样。

这个例子可以从 RUNTIME 子目录中找到。

这个例子中，当按下按钮后，一个从 CUSTOMER.DB 数据表接收公司列表的报表将被创建。这个按钮的 OnClick 事件处理的源代码显示如下：

```
procedure TForm1.Button1Click(Sender: TObject);
var
  v: TfrView;
  b: TfrBandView;
  Page: TfrPage;
begin
  frReport1.Pages.Clear;
  frReport1.Pages.Add;             // create page
  Page := frReport1.Pages[0];

  b := TfrBandView(frCreateObject(gtBand, ''));  // create MasterData band
  b.SetBounds(0, 20, 0, 20);
  b.BandType := btMasterData;
  b.Dataset := 'frDBDataSet1';
  Page.Objects.Add(b);

  v := frCreateObject(gtMemo, '');  // create data field
  v.SetBounds(20, 20, 200, 16);
  v.Memo.Add('[Table1."Company"]');
  Page.Objects.Add(v);

  frReport1.ShowReport;
end;
```

首先这个代码删除报表中所有可用的页面，然后创建一个空白页：

```
frReport1.Pages.Clear;
frReport1.Pages.Add;             // create page
```

接着在这个页上创建"主项数据"区域，并指定数据源：

```
Page := frReport1.Pages[0];

b := TfrBandView(frCreateObject(gtBand, ''));  // create MasterData band
b.SetBounds(0, 20, 0, 20);
b.BandType := btMasterData;
```

```
b.Dataset := 'frDBDataSet1';
Page.Objects.Add(b);
```

下一步是创建一个"Text"对象，并指定到 CUSTOMER.DB 数据表的 COMPANY 字段：

```
v := frCreateObject(gtMemo, '');  // create data field
v.SetBounds(20, 20, 200, 16);
v.Memo.Add('[Table1."Company"]');
Page.Objects.Add(v);
```

最后使用事件处理在预览中显示准备好的报表。

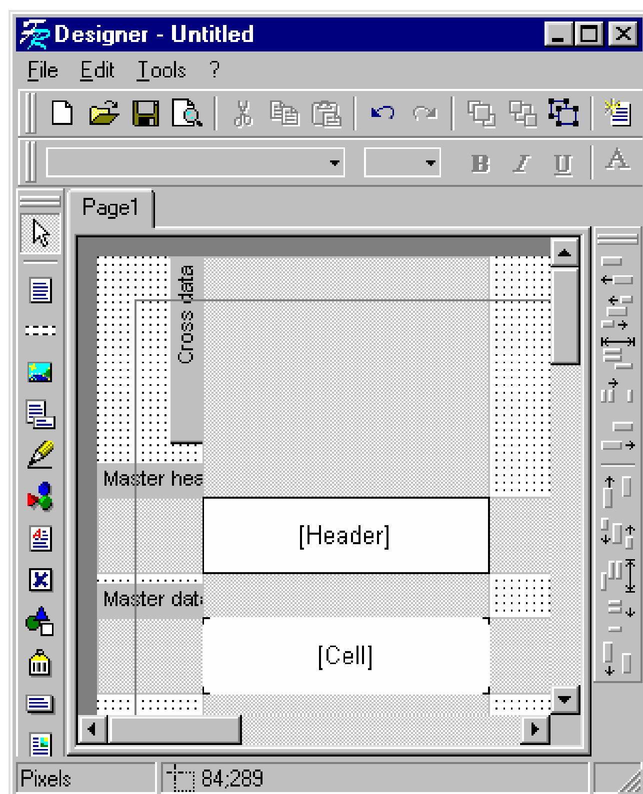
打印可变栏数或未知栏数的报表

实际应用中常常需要打印一个不确定栏数的分栏报表或栏数在执行阶段会改变。例如在开发一个可能需要打印一个数据表数据的数据库编辑器中就需要它。这种情况报表只能使用程序代码创建，但 FastReport 提供了一种更简单的方法来解决这个难题。

就是使用交叉表来达到这个目的。这种报表的与众不同的特点就是在栏中打印数据。在此之前不需要知道分栏数。特别是这种类型的报表交叉部分是垂直放置，而不是水平的。这些区域的"Text"对象被放在水平（例如"主项数据"）和垂直（例如"交叉表"）区域的交叉点上。这种情况报表通过下面的原理创建：对水平区域数据源中的所有记录全部被正确从垂直区域数据源的记录中查找并指定。

因而，对于"主项数据"打印数据源数据表的内容时包含的是数据表自身，各个单独记录的字段列表是"交叉表"区域的数据源。

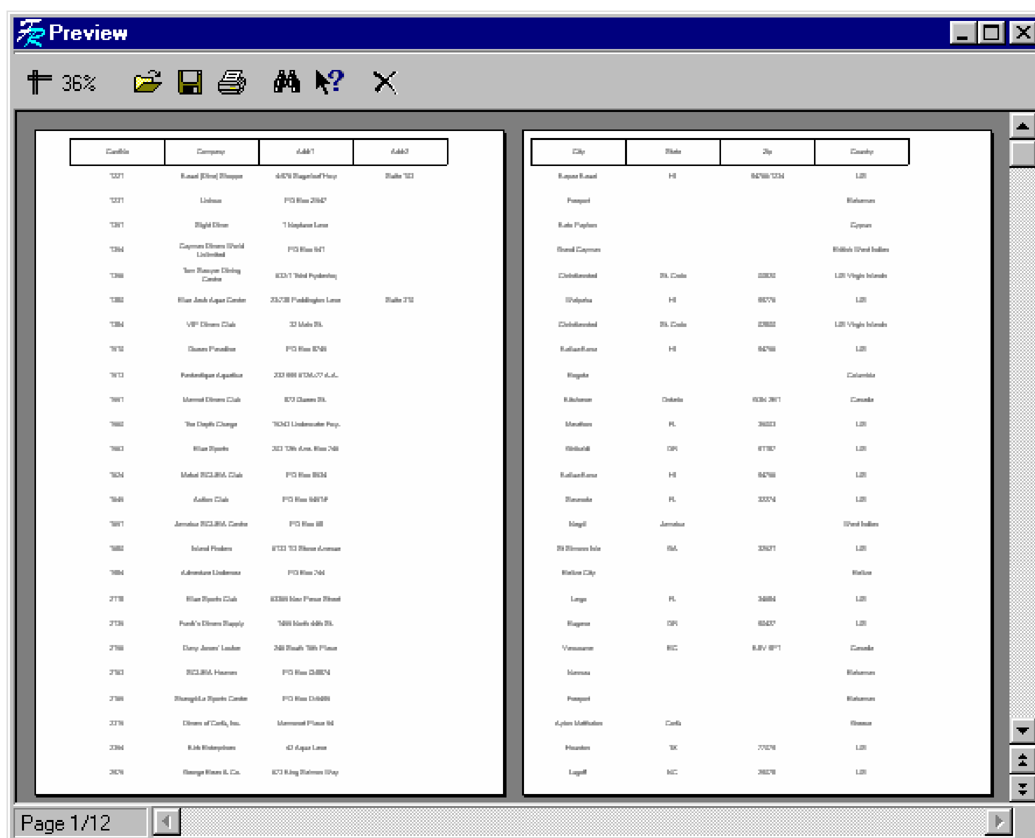
一个交叉表报表的简单例子可以从 PRNTBL1 子目录中找到。从这个例子中看到的报表在这个图象中显示。



这个图象显示“Text”对象放置在“主项数据”和“交叉表”区域的交叉点上，分别属于“主项数据”和“交叉表”区域。如果报表使用 CUSTOMER.DB 中的记录，就将它作为“主项数据”的数据源。报表中的栏目数由两个 TfrUserDataset 虚拟数据源，将 CUSTOMER.DB 的字段数与“记录”数连接。字段的值在 TfrReport 对象的 OnGetValue 事件中设定：

```
procedure TForm1.frReport1GetValue(const ParName: String; var ParValue: Variant);
begin
  if ParName = 'Cell' then
    ParValue := Table1.Fields[frUserDataset1.RecNo].Value;
  if ParName = 'Header' then
    ParValue := Table1.Fields[frUserDataset2.RecNo].FieldName;
end;
```

当这个例子执行时，CUSTOMER.DB 的所有记录将被预览。下面就是预览看到的图象：



打印可变栏数和栏宽可变的报表

在上面的例子中报表的所有栏宽是相等的。可是，在大多数情况下它是需要根据数据确定每栏的宽度。因此，例如，有两个栏，一个栏包含行号而另一个包含 memo 中的字段，显然第一个栏比另一个包含 memo 字段的栏的宽度要窄很多。

扩展的源代码放在 PRNTBL2 子目录中。这个例子除了栏宽是根据数据决定以外，其它的与上一例子相同。这个例子中有两个 TfrReport 的事件处理程序用于动态改变栏宽。

要根据数据动态定义栏宽，就必须使用 TfrReport 对象的 OnPrintColumn 事件。在这个例子中栏中文本框的宽度由字段的“W”字母宽度的个数决定的；对于“日期和时间”字段，设置宽度为

15 个字母"W"的宽度；其它所有的栏宽设为 64 点。

这个 OnPrintColumn 事件处理的源代码显示如下：

```
procedure TForm1.frReport1PrintColumn(ColNo: Integer; var Width: Integer);
var
  Field: TField;
begin
  Field := Table1.Fields[ColNo – 1];
  if Field is TStringField then
    Width := Field.Size * Canvas.TextWidth('W')
  else if Field is TDateTimeField then
    Width := 15 * Canvas.TextWidth('W')
  else
    Width := 64;
  FWidth := Width;
end;
```

OnBeforePrint 事件处理中将栏宽设为上面事件处理中定义的宽度：

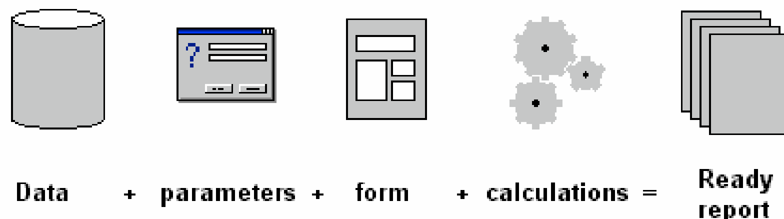
```
procedure TForm1.frReport1EnterRect(Memo: TStringList; View: TfrView);
begin
  View.dx := FWidth;
end;
```

在这个例子中由于每个栏的宽度按数据确定，所以报表打印中非常有效地生成了页面。TfrPrintTable 和 TfrPrintGrid 控件在打印 TDBGrid 和控件内容时就是这样工作的。

最终用户特性

介绍

在“绪论”部分中曾提到，主要的报表创建过程是：



在报表中，在项目中创建，所有过程使用 Delphi 完成。因而，数据访问一般使用数据感知控件：Ttable，Tquery 和类似（作为一种惯例，有一个公共数据库用于所有的项目，并且可以访问它的所有数据表）；对于需要的参数，你可以使用对话框，为项目创建过程；生成过程使用事件处理精确控制。

不考虑效率的情况下，这个方法（应用于大多数的报表生成）并不通用。在某个过程中的一个小改动就需要重新编译整个项目。另外，如果有一个信息系统需要大量报表，这将会引起项目中的对话框的混乱。因此，这个过程在项目中如果需要创建新的报表类型或修改旧的报表时并不有效。

许多报表生成器（象 QuickReport，ReportBuilder，FastReport）慢慢变成独立于项目之外的工具。因此，最重要的是在运行时能够修改报表。然后，是最终用户能够决定用于生成报表的数据源。ReportBuilder4.0 或更高版本，使用类似于 Pascal 的语言帮助你创建用于报表生成的数据过程。FastReport 使用对话框近控件来创建。

因此，FastReport 可以让你：

- 创建或修改报表；
- 创建数据源，用于报表的生成；
- 为用户创建输入报表参数的对话框；
- 使用内嵌的 Pascal 脚本语言处理报表数据和管理对话框；

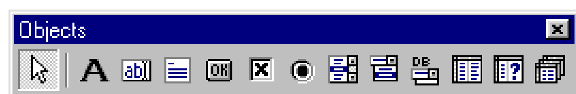
从而可知，它可以帮你创建独立于项目的报表。控件的方式基于 Delphi 的方式。

对话框

开发对话框使用的方法与设计报表窗体的方法相同。设计器工具栏的按钮用于创建一个新的对话框。对象工具栏将自动适应当前窗体类型：报表页面窗体或对话框。

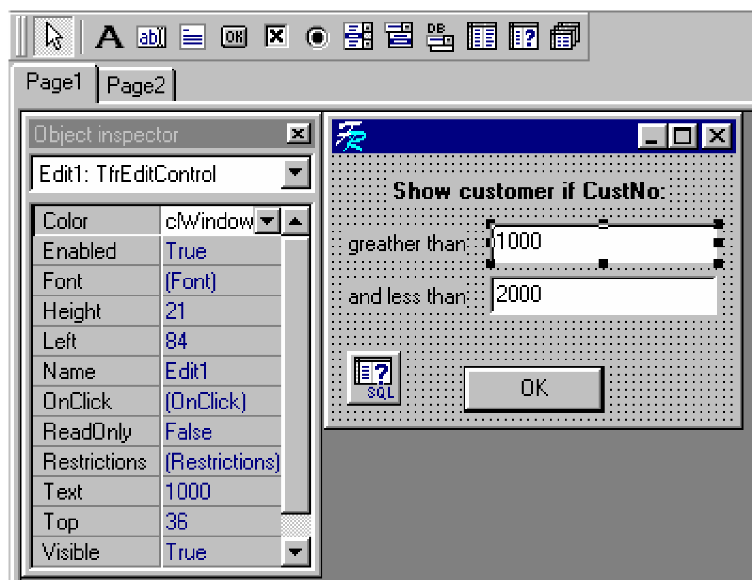


注：这是当“报表页面”窗体激活时的对象工具栏。



注：这是当"对话框"窗体激活时的对象工具栏。

当一个对话框窗体被激活时，设计器的控件将从报表页面类型的控件转为对话框类型的控件。这些控件可以放在窗体上用于创建对话框：



注意创建报表对话框窗体的过程与在 Delphi 中 IDE 中创建对话框窗体非常相似。可以在每个报表中创建多个对话框。这些对话框窗体将按创建它们的顺序执行。所有的对话框窗体都将被执行并可以按"OK"的按钮确认。然后主报表将被生成并报告。

对话框窗体控件

图标	名称	描　　述
	Label	静态文本控件
	Edit	单行编辑栏控件
	Memo	多行编辑器控件
	Button	命令按钮控件
	CheckBox	逻辑选择按钮控件
	RadioButton	互斥选择按钮控件
	ListBox	固定大小的滚动列表控件
	ComboBox	带输入的下拉列表控件

在这里需要着重提一下的是 FastReport 中的所有对话框控件都是从 TfrStdControl 类继承下来的，并且有下列的公共属性和方法：

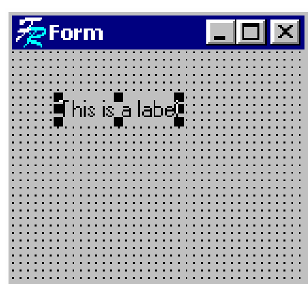
属性	缺省值	描　　述
Color	clBtnFace	对象的背景颜色。
Enabled	True	激活或未激活对象。

Font	-	设置对象文本的字体。
Height	-	设置对象的垂直高度。
Left	-	设置对象的左边坐标。
Name	-	设置对象可以在代码中使用的名称。
OnClick	-	当在对象上单击时触发，相似于 Delphi 的 OnClick 事件。其中可以输入 FastReport 的 Pascal 代码。
Restrictions	-	设置用户修改对象时，控件行为的限制标志集（移动，删除，编辑等）。
Top	-	设置对象的顶边的坐标。
Visible	True	确定对象是否在屏幕上显示。
Width	-	设置对象的宽度。

下面我们来深入了解一下每个控件：

Label

这个控件用于显示静态文本，用户不能修改文本。一般用来显示其它控件的说明或大量的用户提示。

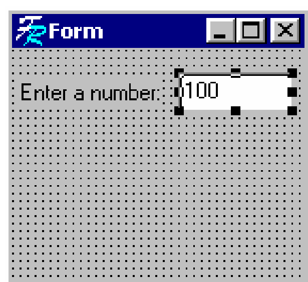


Label 控件有下列属性（还有上面的公共属性）：

属性	缺省值	描　述
Alignment	taLeftJustify	控件 Label 内文本水平对齐位置。
AutoSize	True	确定 Label 的宽度是否根据文本内容自动调整。
Caption	-	设置 Label 显示的文本字符。
WordWrap	False	确定当 Label 文本超出 Label 的宽度时是否换行。当 WordWrap 为"True"时，AutoSize 属性必须设为"False"。

Edit

这个控件用于显示并通过键盘编辑单行文本。



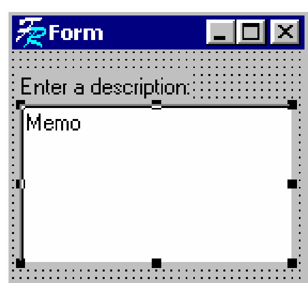
这个控件有下列属性：

属性	缺省值	描　　述
ReadOnly	False	确定用户是否可以编辑文本内容。
Text	-	包含作为显示文本的字符，将被用于控件的缺省显示。



Memo

这个控件用于显示并通过键盘编辑多行文本。



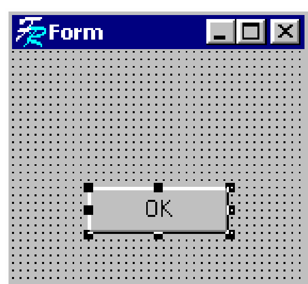
这个控件有下列属性：

属性	缺省值	描　　述
Lines	-	包含作为显示文本行的字符。这些行将被用于缺省显示。
ReadOnly	False	确定用户是否可以编辑文本内容。



Button

这个控件用于显示一个命令控制按钮。



这个控件有下列属性：

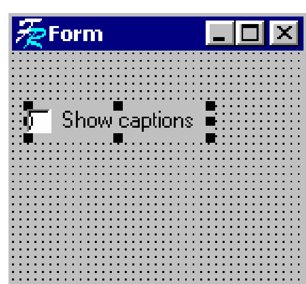
属性	缺省值	描　　述
Caption	-	包含显示在按钮上的文本。
ModalResult	mrNone	确定父对话框的按钮关闭方式（modal），以及返回值。

设置 ModalResult 属性是在点击按钮后关闭模式对话框窗体的一个简单方法。当一个按钮点击后，它的父窗体的 ModalResult 属性设为与按钮的 ModalResult 属性相同的值。通常一个"确认"按钮设为 ModalResult = mrOK，"取消"按钮设为 ModalResult=mrCancel。



CheckBox

这个控件用于显示一个逻辑型的选项按钮。用户可以点击按钮来选中选项，或不再次点击来取消选项。例如 Yes/No，On/Off，True/False，等等。



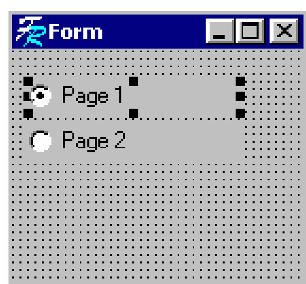
这个控件有下列属性：

属性	缺省值	描　　述
Alignment	taRightJustify	控制标题的位置。
Caption	-	包含显示的标题内容。
Checked	False	指出选项是否被选中。



RadioButton

这个控件使用按钮来表现一组互斥的选项，因此，任何时间一组按钮中只能有一个按钮被选中。在组中的一个按钮点击将选中这个按钮而取消组中的其它。



这个控件的下列属性：

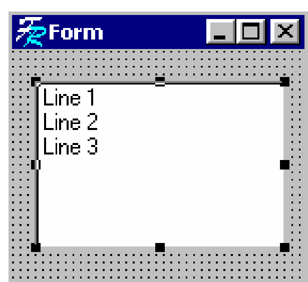
属性	缺省值	描　　述
Alignment	taRightJustify	控制标题的位置。

Caption	-	包含显示的标题上内容。
Checked	False	指出选项是否被选中。



ListBox

这个控件用于显示一个能够从中选择的项目滚动列表。



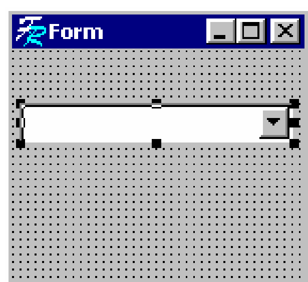
这个控件的下列属性：

属性	缺省值	描　　述
Items	-	包含需要显示的单个项目的文本内容。这些项目在控件中缺省显示。



ComboBox

这个控件用于显示一个下拉的项目滚动列表，用户可以从弹出的一个列表框中进行选择。并联接到一个编辑框控件，它允许用户可以输入项目而不是从下拉列表中选择。



这个控件有下列属性：

属性	缺省值	描　　述
Items	-	包含需要显示的单个项目的文本内容。这些项目在控件的下拉列表中缺省显示。
Style	csDropDown	设置 combo box 的显示类型。
Text	-	在控件的编辑框中显示选择的文本。

ComboBox 的 Style 属性有三个值：

CsDropDown：创建一个可以编辑的下拉列表。用户可以手工输入项目而不需要包含在列表中。所有项目的字符高度相同。

CsDropDownList：创建一个不可编辑的下拉列表。用户不可以手工输入项目，并且编辑框中显

示的项目值必须包含在列表中。所有项目的高度都相同。

CsLookup：列表中的字符串有两部分：String1，String2。Items 属性作为下拉列表中显示的内容，将显示在字符串的第一部分：String1。当一个 String1 值从列表中选中时，包含在 String2 中的值将被显示在编辑框中。例如，控件的弹出项目属性的字符串就象这样：January，1；February，2；等等；可以表现年中的月份。选中的值，String2，将被显示在编辑框的 Text 属性中。

对话框窗体也有一组属性。这些属性当在窗体的空白区域中点击时显示在对象观察器中而非控件。

属性	缺省值	描　　述
BorderStyle	bsDialog	对话框窗体的类型，可变尺寸或固定大小的窗体。
Caption	-	窗口的标题栏中显示的文本。
Color	clBtnFace	设置窗口的背景颜色。
Height	-	设置窗口的垂直高度。
Left	-	设置窗口的左边坐标。
OnActivate	-	在窗口初始化后触发。
Position	poScreenCenter	窗口出现的大小和位置。
Top	-	设置窗口顶边的 Y 坐标。
Type	ptDialog	设置窗口的，对话框窗体或报表页面窗体。
Width	-	设置窗口的水平宽度。

OnActivate 属性与 Delphi 窗体的相同事件类似。它在对话框窗体的所有控件初始化之后，在窗口在屏幕上显示之前执行。可以使用 OnActivate 属性来初始化窗口的控件。例如，ListBox 控件列表的 Items 属性可以在这时设置值。

输入信息到报表中

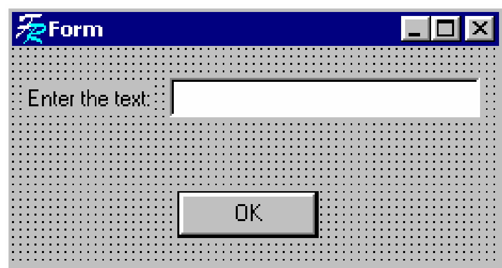
常常需要更改控件中的变量值到报表中。有两种方法：

1．使用变量；

2．直接调用。

使用变量

变量可以用于从控件中改变信息到报表中。例如，需要显示一个编辑框中的文本到报表的标题。

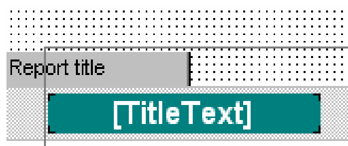


这可以在 OK 按钮的 OnClick 事件中处理：

```
begin
```

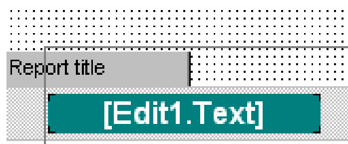
```
 TitleText := Edit1.Text;
end
```

在报表标题，需要在"Text"对象中包含[TitleText]，显示如下：



直接调用

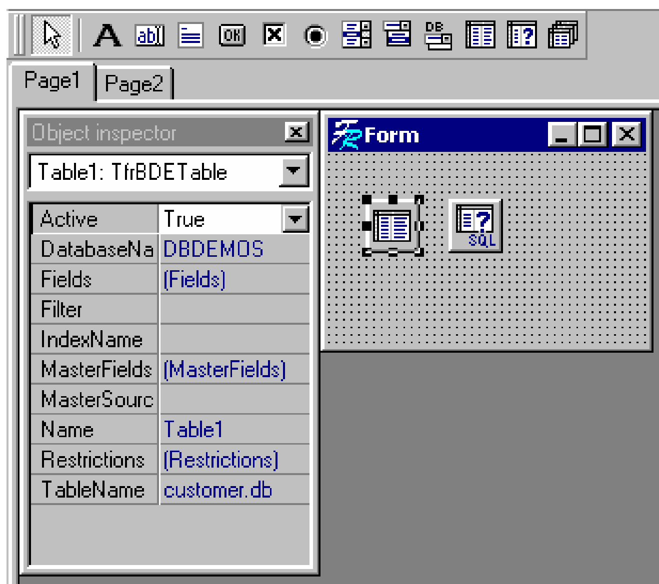
在大多数情况中，直接使用控件的元素往报表中输入值是比较容易的。这种情况，不需要中间变量：



数据访问控件

多数报表，作为惯例，一般从数据库（DB）中访问数据。对于访问这样的数据，Delphi 提供了有效的机制，同样也可用于 FastReport 中。这些数据访问控件中 Ttable 和 Tquery，可以为报表提供数据源。一般情况下，使用多个基于 TdataSet 的控件做到这一点。

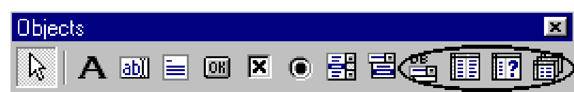
对于数据访问，除了在设计阶段创建，FastReport 也允许在运行中创建新的控件。在 FastReport 中，创建数据访问控件的过程类似于在 Delphi 环境中设计数据库应用。在窗体上放置一个需要的数据访问控件，然后在对象观察器中设置其属性。



FastReport 数据感知控件简介

我们将会讨论使用 BDE 的数据访问控件。项目中的这些控件连接到 FastReport 控件面板中

的 TfrBDEComponents 控件’它们是 TfrBDELookupComboBox ’TfrBDETable ’TfrBDEQuery ’TfrBDEDataBase °



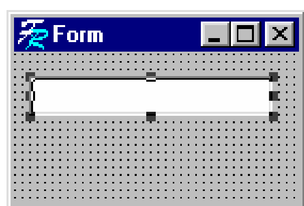
这些控件与数据库控件 TDBLookupComboBox ’Ttable ’Tquery 和 TdataBase 类似 °

图标	名称	描　　述
	TfrBDELookupComboBox	用于从下拉列表中选择其它数据集的需要的字段数据 °
	TfrBDETable	从 Table 中访问数据 °
	TfrBDEQuery	从 SQL 查询中访问数据 °
	TfrBDEDataBase	连接到一个数据库 °

让我们来了解一下每个控件 °

 TfrBDELookupComboBox

TfrBDELookupComboBox 用于为用户提供从下拉列表中选择其它数据集的需要的字段数据 °



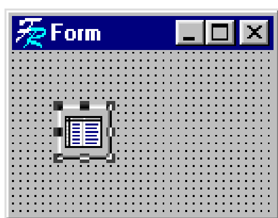
这个控件的下列属性：

属性	缺省值	描　　述
KeyField	-	字段	选择值的检验者 °
ListField	-	在下拉列表中显示值的字段 °
ListSource	-	下拉列表显示数据的数据源 °
Text	-	下位列表中当前选择的值 °

要连接 ComboBox 中其它数据集需要数据值需要三个属性：KeyField ’ListField 和 ListSource °选择的值从 Text 属性中访问 °

 TfrBDETable

这个控件用于从 Table 中访问数据 °



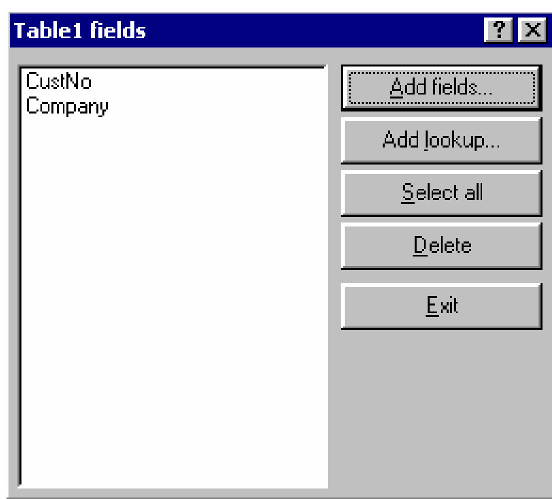
这个控件有下列属性：

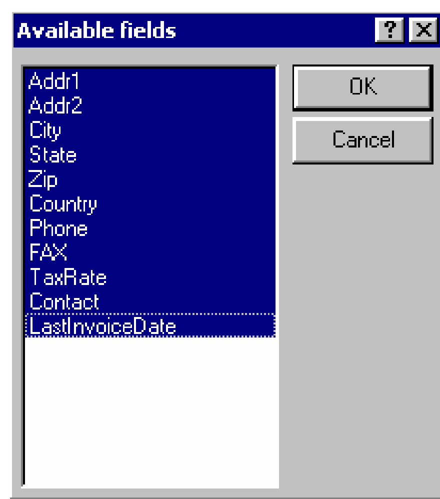
属性	缺省值	描　述
Active	False	确定 table 是否激活。
DatabaseName	-	别名或数据库的别名。
Fields	-	可用的字段列表。
Filter	-	只有满足条件的记录被激活，激活或可见。
IndexName	-	次要索引名。
MasterFields	-	这个字段用于联接到主数据集。
MasterSource	-	主数据集。
TableName	-	Table 名。

上面的属性类似于 Delphi 的 Ttable 控件。要连接控件到 Table，必须设置 DatabaseName 和 TableName 属性。Table 的打开通过设置 TfrBDETable 的 Active 属性为 True。

属性 Fields 和 MasterFields 可通过编辑帮助调整，只需要在对象观察器中按...按钮。

这个 Fields 属性编辑器允许从当前可用的数据表中选择字段。





要增加字段到列表中，点击“添加字段”按钮。一个窗口将显示所有的字段，从中选择字段并按“确认”按钮。要选择多个字段或一组字段，先点击这组字段的第一个字段，然后按住“Shift”键点击最后一个字段。要从列表中删除已选中的字段，选中字段并按“删除”按钮。

要添加查找字段，使用“添加查找字段”按钮。

假设有两个数据表：数据表“Orders”包含字段 N，Date，ClientID 和 Amount；数据表“Clients”包含字段 ID，Name 和 Address。数据表“Orders”包含订单信息（编号，订单日期，顾客编号用于

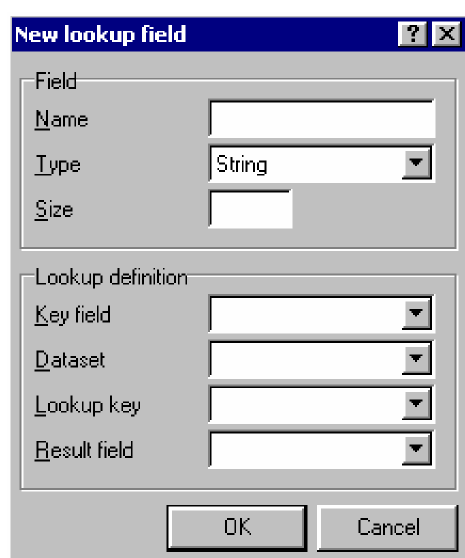
了解谁发出订单和订单汇总）。数据表“Clients”包含了顾客信息（顾客编号，姓名，地址）。

要创建连接“Orders”的空报表：

Order Number - Date - Name of the client – Sum

这需要按字段 ClientID->ID 联接两个表。这可以创建一个查找字段，加入到数据表“Order”中用于按字段名联接到数据表“Clients”。

字段编辑器中的对话框用于创建一个查找字段。

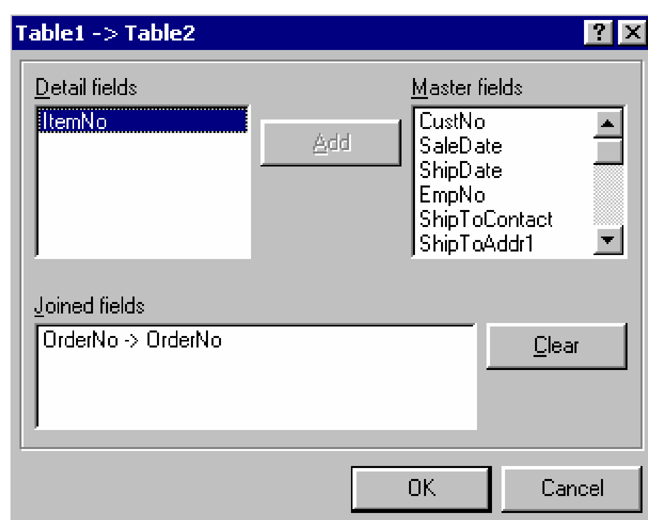


要创建查找字段需要设置字段名和字段类型，还有大小（在这个例子中字段是字符型）。另外，还需要设置下列字段：

- 主关键字段：源数据集中的字段，从查找数据集中按这个字段联接。这个例子中设为 ClientID 字段。
- 数据源：查找数据集。
- 查找关键字：查找数据集的关键字段。这个例子中是 ID 字段。
- 需要替代的返回字段：查找数据集中的字段 根据源数据集中的字段 这个例子中是 Name 字段。

然后，在数据表“Orders”中，就有一个外部字段，内容是 ClientID 字段。你可以访问这个字段，就象其它公共字段一样，但它是只读的。

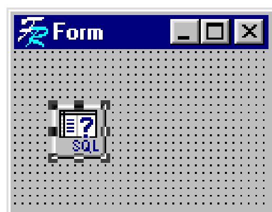
在“MasterFields”属性编辑器中，你可以可视化地联接主从数据集，就象在 Delphi 的 IDE 中一样。



当数据集按主从表设置联接后，主表移动记录，从表即按关键字过滤。由仅与主数据集相关的记录组成。对象数据集联接的数据集字段从左边的列表中选择︵从表︶，然后从右边选择字段︵主表︶，并按﹁添加﹂按钮。这样联接字段移到下面的列表中。要清空下面的列表按﹁清除﹂按钮。已联接的字段与主关键字类型一致。

 TfrBDEQuery

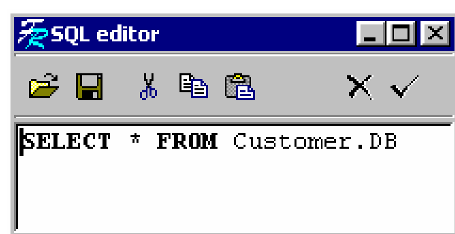
这个控件用于在数据库中执行 SQL 查询。



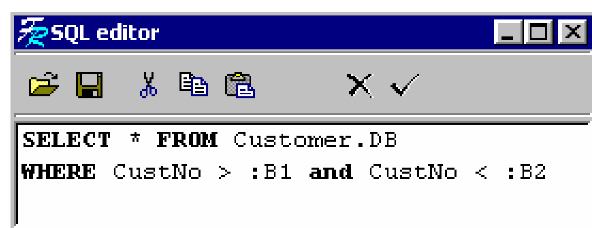
这个控件有下列属性：

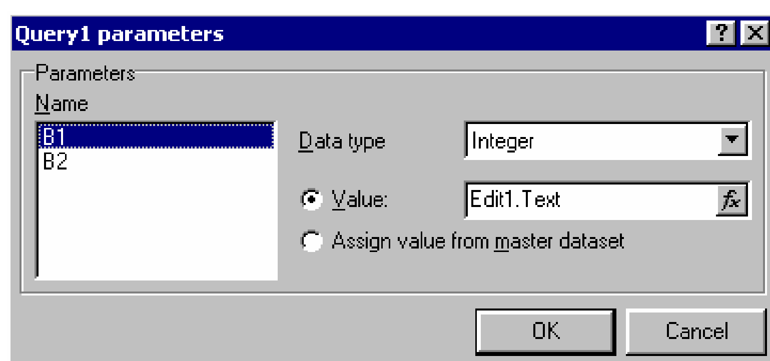
属性	缺省值	描　　述
Active	False	确定查询是否激活。
DatabaseName	-	别名或数据库的别名。
DataSource	-	主数据集。
Fields	-	可用字段列表。
Filter	-	只有满足条件的记录被激活，激活或可见。
Params	-	SQL 语句中的参数列表。
SQL	-	SQL 语句。

属性 Active，DatabaseName，Fields 和 Filter 与上面的 TfrBDETable 控件属性类似。属性 SQL 有编辑器一个用于输入 SQL 语句。



当 SQL 语句包含参数时，属性 Params 也有一个编辑器用于进行编辑。



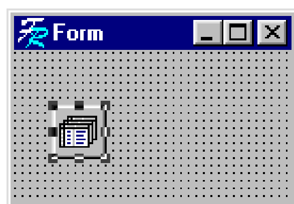


在这个对话框中，可用于设置各个参数的类型，以及配置参数值的来源，来自主数据集或指定一个具体的值。现在这个例子中参数值来自主数据集，主数据集包含与参数名相同的字段。另外这个字段必须包含在可用的字段列表中（请参考字段编辑器）。因此，它可以表明参数类型。



TfrBDEDataBase

这个控件用于连接到一个数据库。它与 Delphi 中的 TdataBase 控件类似。

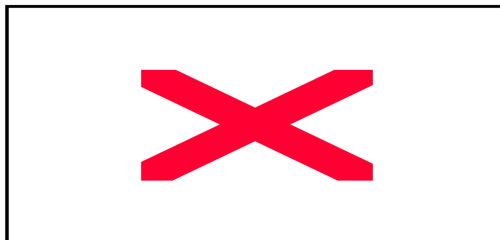


属性	缺省值	描　述
AliasName	-	别名或仅在该项目使用的数据库别名。
Connected	False	如果为 True，则激活连接。
DatabaseName	-	增加到别名列表中的别名。
DriverName	-	连接到的数据库的驱动程序名。
LoginPrompt	True	确定在连接到数据库时是否提示用户输入口令。

Params	-	连接参数。

这个控件允许我们连接到数据库（通常用于连接到数据库服务器）。

与在 Delphi 中一样，必须设置 AliasName 和 DriverName 属性。要设置连接属性，在对象观察器中选择“Params”属性并运行属性编辑器。



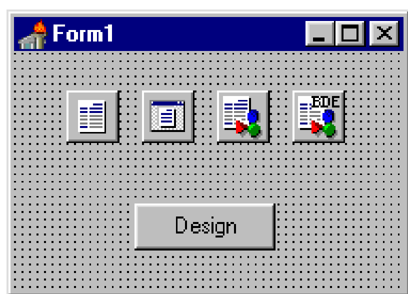
当“LoginPrompt”属性为 True 时，在你连接到数据库时，数据库登录对话框将出现。如果你在数据库参数中提供用户名和口令，就象下面这样，你需要设置这个属性为 false。

```
SERVER NAME=Path_to_gdb_file
USER NAME=SYSDBA
PASSWORD=masterkey
```

创建报表

让我们来看一个怎样创建一个使用数据访问控件的简单报表。我们将使用 Delphi 中的 DBDEMOS 示例数据库作为数据源。

首先，我们为我们的程序新建一个项目。创建一个新窗体，并在窗体上放置 TfrReport，TfrDesigner，TfrDialogControls 和 TfrBDEComponents 控件。



为“Design”按钮定义如下事件处理：

```
procedure TForm1.Button1Click(Sender: TObject);
begin
  frReport1.DesignReport;
end;
```

然后编译并运行这个项目。你就可以使用最终用户报表设计器了。

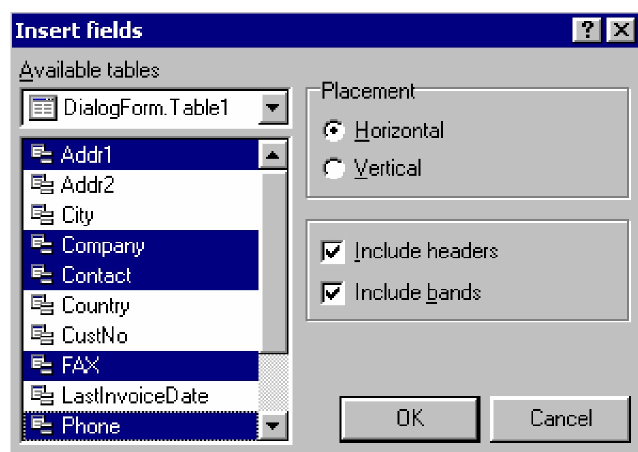
当你点击“Design”的按钮时，设计器将调入一个空白的报表。让我们来考虑一下怎样在这个环境中创建一个简单报表。

简单的“表格式”报表

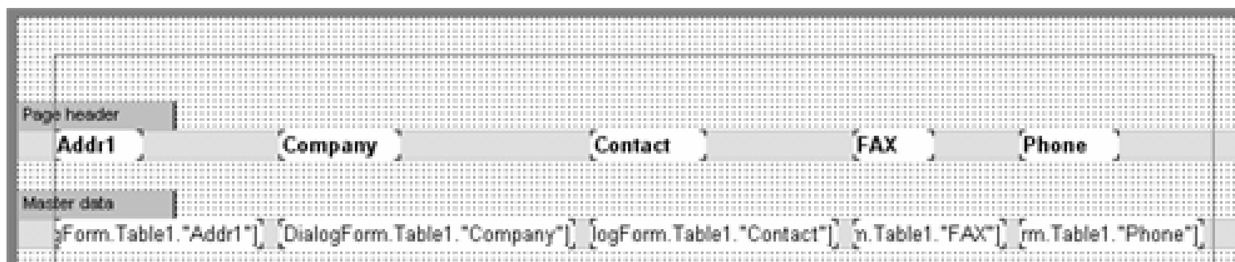
这个报表从数据库的一个表显示数据。通过下列步骤创建这个报表：

- 添加一个对话框窗体到报表。这个窗体用于控制数据访问控件。
- 放一个 TfrBDETable 控件到窗体上，并改为下面的设置：DatabaseName = 'DBDEMOS', TableName = 'Customer.db'。

- 切换到报表窗体，在设计器中向导工具栏中点击"插入字段到报表"按钮 。在打开的对话框中选择需要的字段并按"确认"按钮。



现在报表看起来就象这样：

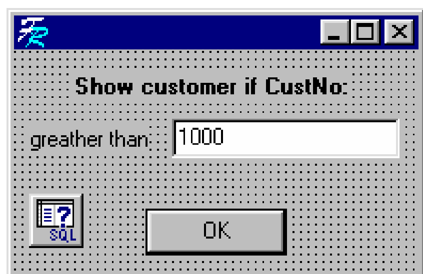


在工具栏中按预览按钮 观察报表。

带参数的报表

让我们再来看一下一些复杂报表的创建，需要在生成报表之前在对话框中输入参数，步骤如下：

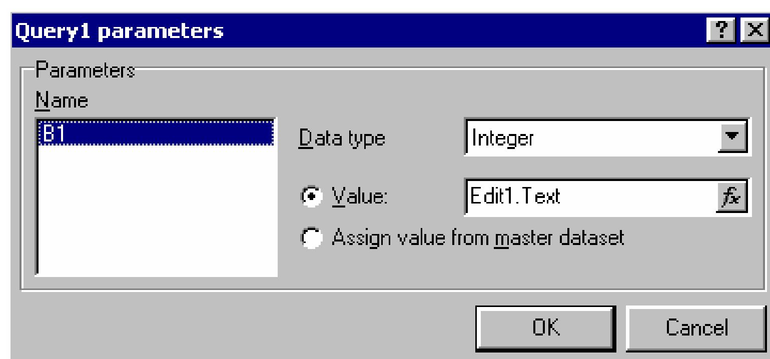
- 添加对话框窗体到报表中。
- 放置 Query，Label，Edit 和 Button 控件到窗体上。



- Query 控件设置如下：DatabaseName = 'DBDEMOS', SQL = 'Select * from Customer.db where

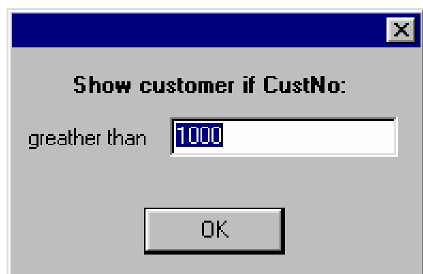
CustNo > :B1' °

- 打开 Query 控件的 Params 属性编辑器并修改参数属性 °



- 切换到报表窗体并按设计器向导工具栏中的“插入字段到报表中”按钮 °在打开的对话框中选择需要的字段并按“确认”°

当报表生成时 ’你可以通过对话框提供输入客户编号 °只有在你输入编号并按“确认”后 ’报表才会继续生成 °所有的客户中只有你输入的编号才会被打印 °

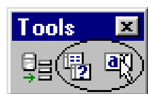


TfrDataStorage 控件

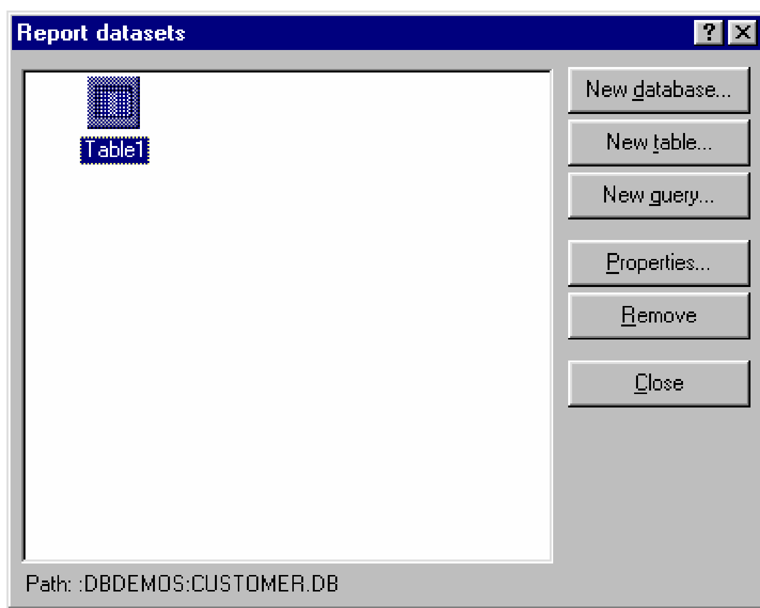
TfrDataStorage 控件用于在运行中创建数据库 ’数据表和查询 ’设置可用的字段列表 ’创建查询字段 ’主从结构 °这个控件允许执行与在 Delphi 的 IDE 中相同的操作 °所有信息将和报表窗体一起保存 °

在有些时候 ’控件已经过时 °必须使用 TfrDialogControls + TfrBDEComponents 进行替换 °

当控件已连接 ’“Data Manager”和“Parameters dialog”被加到设计工具列表中 °



首要的是准备工作数据：定义数据库 ’数据表 ’查询并编辑它们 °其次是确定用于查询的参数 °所有用这个工具创建的数据库 ’数据表和查询都放在“报表数据”的数据模块中 ’这是一个在运行期间可以访问的模块 °对于各个数据表或查询都有相同的 FastReport 数据源 ’也可以是数据表 ’只不过有下划线 °

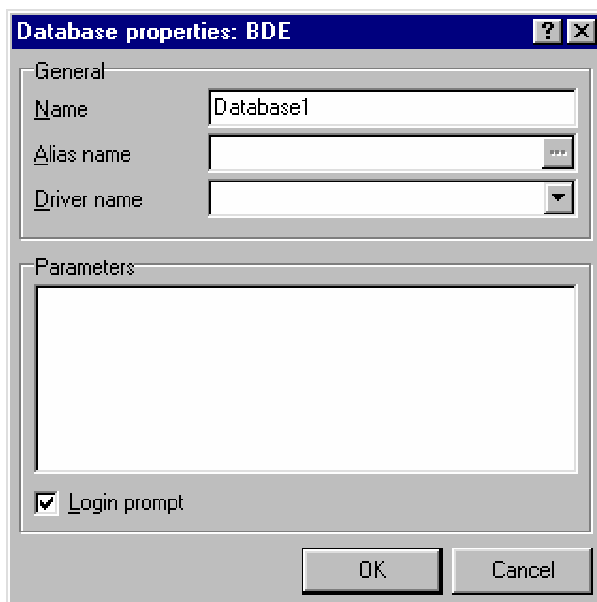


数据管理器支持 BDE，InterbaseExpress 或 Microsoft ActiveX Data Object(ADO)。IBX 和 ADO 不需要使用 BDE。要激活适当的数据库引擎，需要在 FR.INC 文件中加入相应的参数行。

连接到一个数据库

如果你使用 ADO 或 IBX 数据控件需要连接到数据库。在创建任何数据表或查询之前，你必须先创建数据库。

要创建一个数据库，打开"数据管理器"。在出现的窗口中点击"新数据库"按钮。然后将出现"数据库参数"对话框。如果你使用 ADO 或 IBX，设置数据库控件名或选择适当的 DB 文件。如果你使用 BDE，请设置数据库控件名，设置别名，你想添加到别名列表中的，然后选择适当的驱动程序。



你也可以设置 DB 参数，例如用于连接 SQL 数据库的用户名和口令。注意：不同驱动使用不同

的参数名。例如，要连接到 BDE 本地驱动 InterbaseSQL 服务器，使用下面的参数：

```
SERVER NAME=Path to your *.gdb file
USER NAME=SYSDBA
PASSWORD=masterkey
```

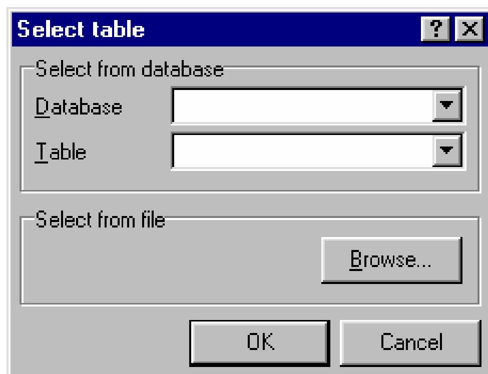
要连接到 IBX 的 InterbaseSQL 服务器，使用下面的参数：

```
user_name=SYSDBA
password=masterkey
```

当"Login prompt"选项打开后，当你连接到 SQL 服务器时将出现数据库登录对话框。如果你在数据库参数中设置用户名和口令，你需要关闭这个选项。这可以自动连接到数据库。

打开一个数据表

要创建一个新数据表，打开"数据管理器"。在出现的窗口中点击"新建表"按钮。然后将出现"选择表"对话框。你可以从可用的数据库中选择需要的数据表或直接从文件中调入（仅用于 BDE 的本地 Paradox 或 Dbase 数据表）。选中数据表后，出现"数据表参数"对话框。在这里可以设计数据表名（缺省是"Table"+3 个数字）。这个表的 TdataSource 和 TfrDBDataSet 一起被创建。TfrDBDataSet 的名字也和数据表一样，只不过加了下划线。



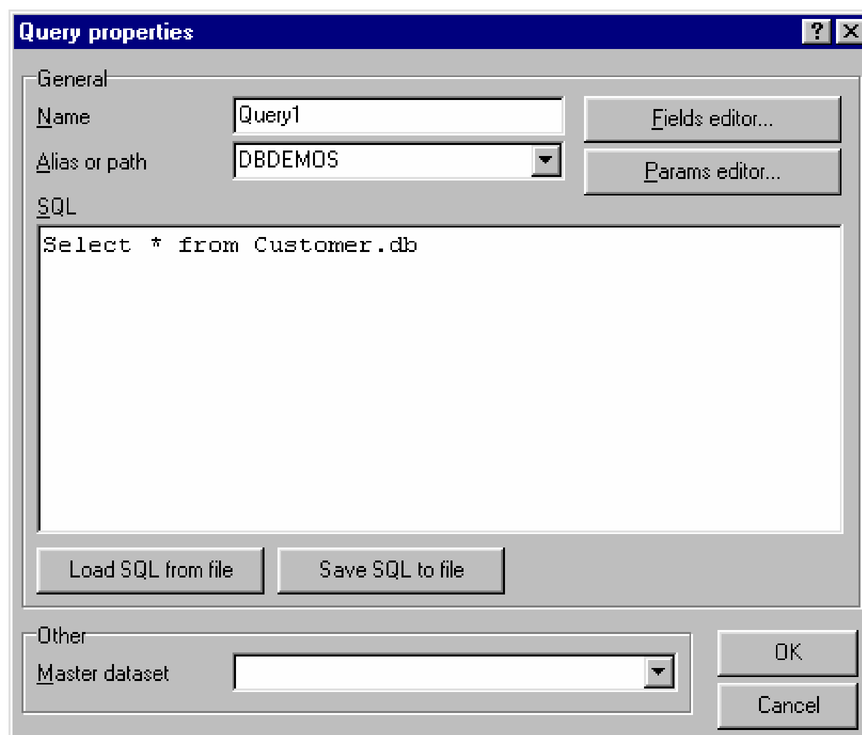


同样在这个窗口中你可以打开字段编辑器，为数据表选择索引（如果数据表的次要索引），设

置过滤表达式。要为数据表设置主数据集，你需要从“主表”列表中选择数据集，并从主数据集的“主表字段”那行中选择字段（或点击右边的按钮，在打开的对话框中可视地选择联接字段）。

生成一个查询

打开工具“数据管理器”创建新的查询。在出现的窗口中点击“新查询”按钮。然后打开“查询属性”对话框。这里需要设置查询名称（缺省是“Query”＋一个数字）。对于查询必须已创建 TdataSource 和 TfrDBDataSet。TfrDBDataSet 的名称与查询一样，只不过加上下划线。



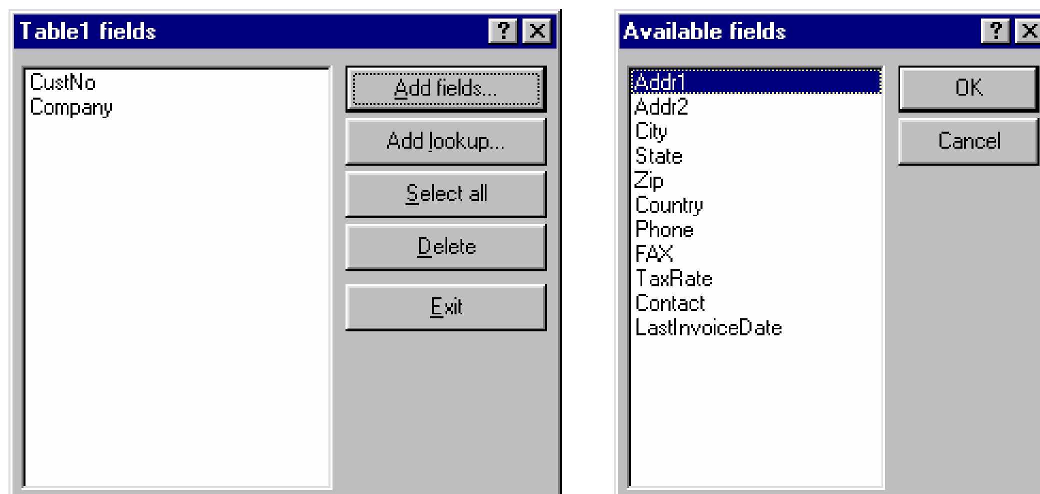
“别名”下拉列表中包含已建好的数据库的路径。这里可以是下列列表中的一个值，也可以是一个存放表的路径。这个字段有可能是空的，这时就需要在查询语句中包含别名或路径。然后就需要在“SQL”文本中输入查询语句。如果这个查询语句中包含参数（名字开头有“：”，各个参数在参数编辑器中描述。

同样，在这个窗口中你也可以执行字段编辑器和设计主数据集。要为查询设置主数据集，你需要从“主项数据”列表中选择需要的数据集。然后查询语句中就可以包含参数：主数据集中的字段名。另外，每个参数都有一个“从主数据源中获取”。

在点击“确认”后，查询执行。如果查询语句或参数的描述错误，那么将出现一个错误提示并弹出查询属性编辑器。

字段编辑器

在这个编辑器中设置数据表或查询中可用的字段列表。如果列表是空的，那么可以使用所有的字段。



点击“增加字段”按钮往这个列表中添加字段。从出现的窗口中选择字段并按“确认”。要选择一组字段，首先用鼠标选中第一个字段，然后按住“Shift”键，点击最后一个。要增加一个查询字段，点击“增加查询字段”按钮。参考“创建查询字段”。

要从列表中删除字段，点击“删除”按钮。

创建查询字段

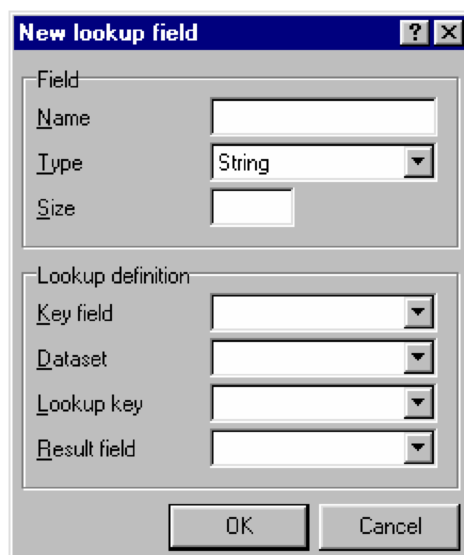
我们通过一个例子来说明查询字段需要些什么。

有两个数据表：数据表“Orders”包含字段 N，Date，ClientID 和 Amount；数据表“Clients”包含字段 ID，Name 和 Address。数据表“Orders”包含订单信息（编号，订单日期，顾客编号用于了解谁发出订单和订单汇总）。数据表“Clients”包含了顾客信息（顾客编号，姓名，地址）。

要创建连接“Orders”的空报表：

Order Number - Date - Name of the client – Sum

这需要按字段 ClientID->ID 联接两个表。这可以创建一个查找字段，加入到数据表“Order”中用于按字段名联接到数据表“Clients”。



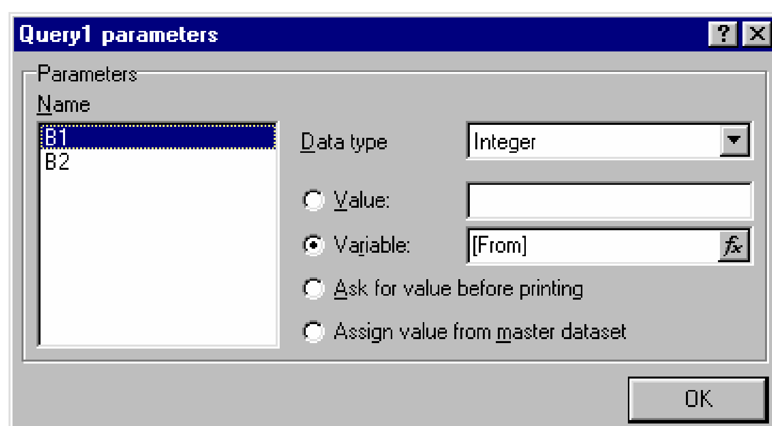
创建查询字段的对话框也可以从字段编辑器中调用。要创建一个查询字段，需要设置名称和类型，同样也有大小（这个例子中是选择字符类型）。另外必须设置下列字段：

- 主关键字段：源数据集中的字段，从查找数据集中按这个字段联接。这个例子中设为 ClientID 字段。
- 数据源：查找数据集。
- 查找关键字：查找数据集的关键字段。这个例子中是 ID 字段。
- 需要替代的返回字段：查找数据集中的字段（根据源数据集中的字段）这个例子中是 Name 字段。

然后，在数据表"Orders"中，就有一个外部字段，内容是 ClientID 字段。你可以访问这个字段，就象其它公共字段一样，但它是只读的。

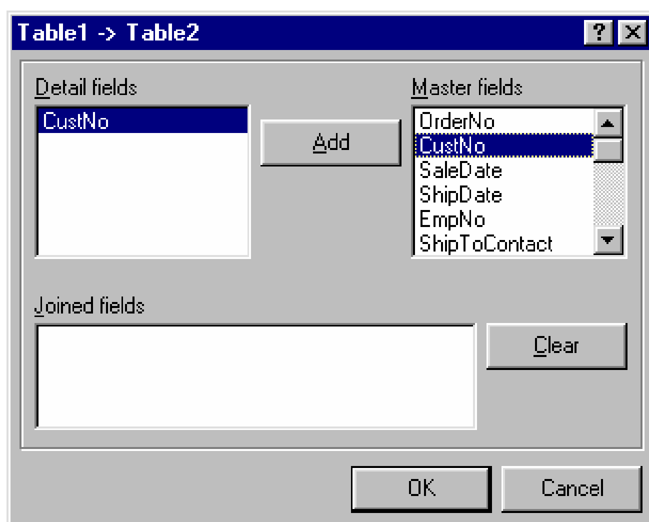
查询参数编辑器

这个对话框可以设置各个参数的类型，同样也可以设置参数值的来源：从主数据集，从对话框或仅仅设置一个具体的值。当参数从主数据集中获取时，数据集包含的字段名可以连接到参数名。另外，这个字段必须在可用的字段列表中（参考"字段编辑器"）因此，它也可以指出参数的类型。



联接数据

这个对话框用于可视化地设置主从数据集的联接字段。



当各个数据集间按主从结构联接时，在主数据集上移动时，将设置细数据集过滤。因而，只包含联接到主数据集的字段。要联接数据字段，从左边列表选择数据（细数据集），然后在右边的列表选择字段（主数据集），点击“增加”按钮。这时，联接字段移到下面的列表。按“清除”按钮清除下面的列表。联接的字段数据类型必须相同并且是主关键字。

参数对话框

如果作为报表数据源的查询包含一个或多个参数，需要选中一个选项“需要值”（参考查询参数编辑器），然后在报表生成时屏幕上将出现参数对话框用于输入值。在这个对话框中将包含所有参数，包括用于生成报表的所有查询。

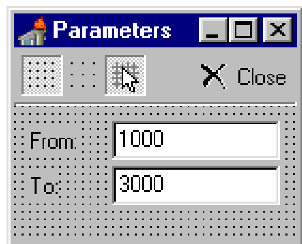
缺省使用编辑框输入参数值，旁边有一个描述——参数名。如果大量参数仅仅包含自身，那么这是非常不方便的——因为参数名是简单的英文说明。这种情况就需要利用“设计器对话框”。

如果有参数的报表激活时，这个对话框就会出现。空白报表或报表不连接到带参数的查询，这个对话框就不会出现。

参数对话框设计器

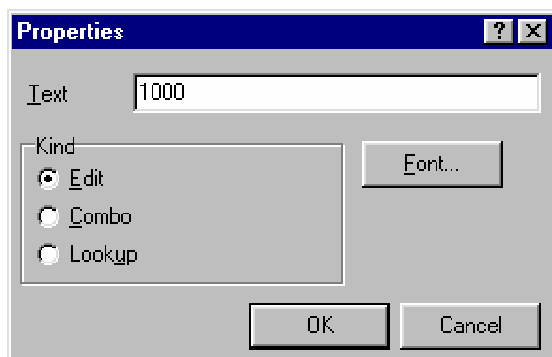
对话框设计器非常适合用于修改出现的参数输入对话框。它可以按需要重排对话框中对话的位置和描述（缺省使用参数名）。另外，设计器允许用其它类型的窗口元素取代缺省的编辑框输入参数值。

但设计器不允许增加新对象或删除对象。它只能用于设置它们的位置和大小，以及设置属性（例如，参数的字体）。所有操作通过鼠标完成。鼠标的操作与报表窗体设计器一样。



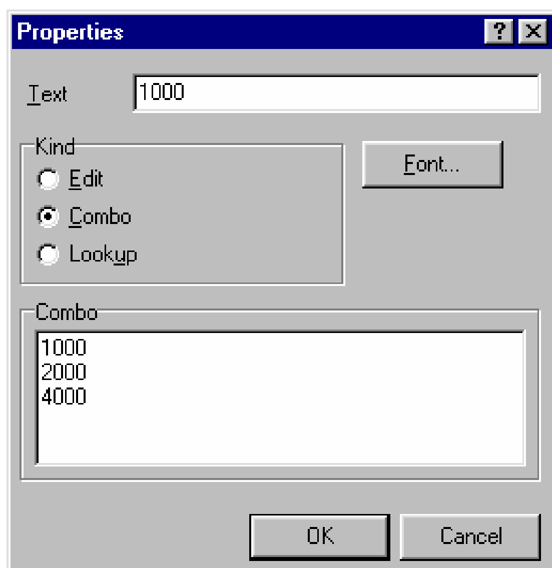
要设计对象属性，只要在它上面双击鼠标。在出现的对话框中，可以设置文本，在对象中显示。

如果选中多个对象，也可以对齐。如果选中对象：窗口元素，可以在属性对话框中重新设置“对象类型”。



可以选择三种类型：

- 编辑框：显示文本行，可以输入任何文本。如果选中在对话框的“文本”字段中输入内容，这个值将出现在输入对话框的编辑框中。
- 列表：显示下拉列表。这是经常使用的，如果需要用户从多个值中选择的话。如果列表中的行包含字符“；”，那么将显示分号前面的部分，选中这个值后分号后面的部分将显示在参数行。这是一个非常方便的方法，例如需要选择月份：列表中的行象“January; 1”，“February; 2”，列表中月份的标题将被显示，而参数中将得到数字。



- 查询列表：用于从参考中选择值（有主关键字的数据表和字段标题）。要设置查询需要的关键字段的值，用于返回到参数中，将被显示在下拉列表中。

也可以设置窗口的大小：只能允许这个改变，执行后对话框下方将出现“确认”按钮。

内置语言

FastReport 内置一种类似于 Pascal 的语言解释器。这个解释器一个强大的独立（Delphi 或 C++Builder）的脚本写作语言，用于创建报表。

这个语言所使用的 Pascal 语言元素如下所列：

- 操作符 赋值操作 条件语句 条件语句和无条件转移 **if...then...else, while...do, repeat...until, for..to..do, goto**；
- 插入语句 **begin...end**；
- 所有类型的变量，数组；
- 通过点号访问 FastReport 对象的属性和方法。

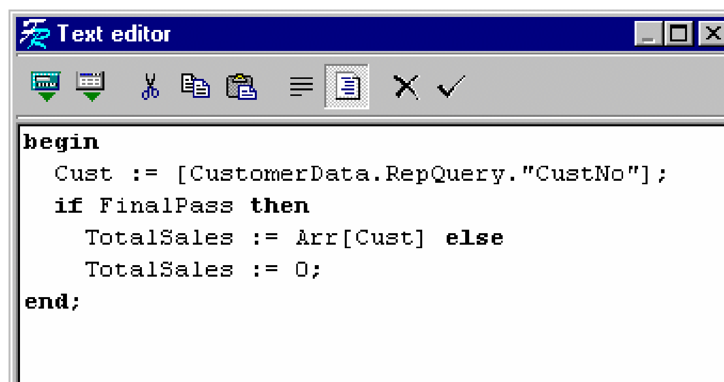
较 Object Pascal 而言，这个语言是比较简单的。下面是简化操作：

- 所有变量是 Variant 类型，不需要定义变量类型；
- 所有变量是全局的，没有局部变量；
- 没有象类，记录，枚举等数据类型；
- 不能写自己的过程或函数；
- 没有循环的中断操作符（break，continue）；
- 过程或函数不能有三个以上的参数；
- 由于事实上没有数据类型，因而不能控制数据类型，当写一个逻辑表达式时就可以看到这点；
- 只能定义一维数组。

不管如何简化，解释都能让你正确进行适当的数据处理。这个脚本语言你可以访问报表对象的所有方法和属性，也能访问数据表字段。在脚本语言中你可以声明变量和数据，应用于所有的报表中 内置语言的能力可以通过一个例子来证明 象在分组头打印分组汇总这样的难度对于 FastReport 来说是非常容易的。

脚本语言和对象

每个对象都有一个或多个代码段。脚本编辑器在文本编辑器窗口中运行（要查看脚本你可以点击窗口顶部的按钮）。这个脚本在打印一个对象之前运行。（这个脚本附属在对象的 OnBeforePrint 属性中）。



不仅只有对象有脚本。Bands 和报表页也可以定义脚本。要定义 band 的脚本，你必须打开 band 的 OnBeforePrint 属性编辑器（从对象观察器中或选中 band 并按 Ctrl+Enter）。要定义报表页面的脚本，你必须打开报表页面的 OnBeforePrint 属性编辑器（在页面的空白区域点击或从对象观察器中

调用编辑器）。对话框窗体和报表页面的脚本附属在 OnActivate 属性中。所有对象的脚本都可以通过访问它们自己的 Memo 属性或按 Ctrl+Enter。

编写代码

在脚本中，你可以使用报表对象的属性和方法，数据表的字段和各种常量。同样你也可以创建变量和数组用于所有的报表。你也可以使用过程和函数。

u 使用变量

这里不需要指定变量类型，它们都是变体类型。你可以使用拉丁字母，数据和下划线作为变量名。脚本中的变量可以在对象中使用，并且变量列表中的变量同样也能用于脚本。脚本变量存储在 **TfrVariables** 中，并可以通过 **TfrVariables** 访问变量。

这有一个使用中间变量的例子：

```
begin
  Cust := [CustomerData.RepQuery."CustNo"];
  if FinalPass then
    TotalSales := Arr[Cust] else
    TotalSales := 0;
end;
```

这个例子中，我们定义了一个 Cust 变量，并设置它等于数据表字段值。

你也可以调用在数据字典中定义的变量，系统变量和用户变量。这种情况，变量名可以正常语法不支持包含符号（例如 Page# 系统变量）。要调用这样的变量你必须使用方括号：

```
begin
  a := [Page#];
end
```

u 使用数据库字段

在你的脚本中可以使用数据表。正确的语法：*[FormName.TalbeName."FieldName"]*。

当数据表和报表在不同的窗体（或数据模块）中时必须使用完整路径。如果控件在同一个窗体中，你只需要使用 *[TableName."FieldName"]* 这样的地址。

如果你使用的数据表在 band 中，并且你在其中使用，你可以这样写 *["FieldName"]*。例如，你有主项数据 band，连接到一个 Customer.db 数据表的数据源，你可能通过短路径在报表中访问数据表的字段。使用完整路径并不会使性能下降，FastReport 在内存中保存数据字段名。

u 数组

除了变量外，你也可以在你的脚本中创建数组。只能使用一维数据，但你可以使用它们来代替二维数组。

例如使用一个数据：

```
begin
  MyArr[0] := 'a'; MyArr[1] := 'b'; MyArr[3] := 'd';
```

```
  MyArr[2] := MyArr[0] + MyArr[1] + 'c' + MyArr[3];
end;
```

事实上数据按 Arr_*array_name_index* 格式存储在 frVariables 列表中。上面的例子这样包含在 frVariables 中：

```
Arr_MyArr_0 := 'a'
Arr_MyArr_1 := 'b'
Arr_MyArr_2 := 'abcd'
Arr_MyArr_3 := 'd'
```

- **常量**

在你的脚本中可以使用常量。一个简单的例子是使用数字，字符和逻辑常量：

```
begin
  a := 0;
  b := 'abcd';
  c := True;
  d := 'That''s all!';
end;
```

需要注意的是在常量字符串内使用单引号（与 Pascal 一样，必须这样定义：`d := 'That''s all!'`。

除了使用简单的常量外，你也可以使用象颜色名，字体名等等。下面是可用的常量列表：

- 颜色：clWhite，clBlack 等等。（所有标准颜色和系统颜色；
- 对话框返回值常量：mrNone，mrOk，mrCancel；
- 系统常量：CRLF，Null；
- 字体修饰类型：fsBold，fsItalic，fsUnderline；
- 对象边框：frftNone，frftRight，frftBottom，frftLeft，frftTop；
- "Text"对象的文本对齐：frtaLeft, frtaRight, frtaCenter, frtaVertical, frtaMiddle, frtaDown；
- band 对齐：baNone, baLeft, baRight, baCenter, baWidth, baBottom。

另外还有对象附带的常量，例如："CheckBoxObject"对象的 *csCheck*。任何一个你在对象观察器的属性下位列表中找到的，你都可以在脚本中作为常量使用。

- **使用对象**

你可以在脚本中使用报表对象的属性和方法。报表对象指可视对象，控件，bands，报表页面和报表自己。要使用一个对象必须使用点号，例如：*Memo1.Text*。使用内部属性和方法是不需要的点号的。

显示在对象观察器中的属性都是可以使用的。一些象字体这样的复合属性必须象这样才能使用：*Font.Name, Font.Size* 等等。

```
begin
```

```
  Memo1.Font.Name := 'Courier New';
  Memo1.Font.Size := 10;
  Memo1.Font.Color := clRed;
  Memo1.Font.Style := fsBold + fsItalic
end;
```

象 TStrings (Memo, SQL, Items 等)的属性可以通过它们的索引引用：

```
if Memo1.Lines[1] = 'a' then
  Memo1.Lines[1] := 'b'
```

这样的属性也可以使用 Add, Delete, Clear 和 Count：

```
if Memo1.Lines.Count > 10 then
  Memo1.Lines.Delete(10)
else
begin
  Memo1.Lines.Clear;
  Memo1.Lines.Add('a');
end;
```

对象的属性和方法的完整列表可以从"对象属性和方法"一节中找到。注意，使用一个不存在的方法或属性将提示错误，写代码必须小心。

- **使用过程和方法**

脚本可以包含过程和方法的调用。解释器的一个独特的属性，或者说是对过程或方法处理更精确的解析，过程和函数不能有 3 个以上的参数。脚本可以同时使用在项目中的内置的过程和外部过程。内置过程和函数列表在"内置过程和函数"节中。

注意：当使用一个过程或函数时，不能在过程名和括号间输入空格。

对象修改

在你的脚本中，你可以对对象进行修改，象改变大小，颜色，内容等。*你需要记住在扫描一遍的报表中你不能修改已经创建好的对象*。也就是说如果你试图修改按报表合计 band 中的一个对象修改报表标题中一个对象的内容，将不会改变。然而，如果你设置为扫描两遍报表就可以进行这样的修改。

在多页面报表中也使用相同的方法。你可以通过它们的名称访问所有对象（在报表中的名字是必需的）。但只有未生成的对象是可以修改的。如果你需要在一个项目中修改已生成的，你需要设置扫描两遍报表。

内置函数

合计函数

合计函数可以在报表合计、页脚、主项数据脚、细项数据脚、子细项数据脚、分组脚和交叉表脚 band。

- Sum(<expression> [, band] [,1]).为 band 中的行按<expression>计算值的总和。如果 band 的参数没有设置，sum 缺省是计算所有数据值（主项数据，细项数据和子细项数据）；另外，sum 只能用于已命名的 band 中的数据。如果使用“1”这个参数，sum 也可用于非可视对象。例如：

  Sum([Part total], Band1);

  Sum([[Part total] + [Part price]]);

  Sum([Part total], Band1, 1).

- Avg, Min, Max.语法与 sum 函数类似。函数 Avg 计算算术平均值，函数 Min 返回最小值，函数 Max 返回最大值。
- Count(<band>).返回数据行的总数。例如：Count(Band1)。

字符处理函数

- Str(<value>).将包含的数值转换成字符。
- Copy(<string>, <from>, <count>).按长度<count>返回字符串<stirng>中的子串。
- If(<expression>, <string1>, <string2>).如果表达式为 True，返回字符串<string1>；否则，返回字符串<string2>。
- FormatFloat(<formatstr>, <value>).将浮点数按格式符转换为字符。<formatstr>值在 Delphi 的文档中“Formatting strings”一节中描述。
- FormatDateTime(<formatstr>, <value>).将时期、时间按格式符转换为字符。<formatstr>值在 Delphi 的文档中“Formatting strings”一节中描述。
- StrToDate(<value>).转换字符<value>为日期。
- StrToTime(<value>).转换字符<value>为时间。
- UpperCase(<value>).转换字符<value>为全部大写。
- LowerCase(<value>).转换字符<value>为全部小写。
- NameCase(<value>).转换字符<value>第一个字符为大写，其它字符为小写。
- Length(<string>).返回<string>的长度。
- Trim(<string>).整理（去除）<string>的开始和末尾的所有空格，并返回结果。
- Pos(<substring>, <string>).返回子串<substring>在字符串<stirng>中的位置。

算术运算函数

- Int(<value>).返回数值<value>的整数部分。
- Frac(<value>).返回数值<value>的小数部分。
- Round(<value>).返回四舍五入的数值。
- value1 Mod value2.返回 value1 除以 value2 的余数。
- MinNum(<value1>, <value2>).返回两个<value>的最小值。
- MaxNum(<value1>, <value2>).返回两个<value>的最大值。

其它函数

- Input(<caption> [,<default>]).显示标题为<caption>并有一个编辑框的对话框。如果“defalut”

参数被设置，则将这个字符显示在编辑框的。在用户点击"确认"后，返回输入的字符。

- Date.返回当前的系统日期。
- Time.返回当前的系统时间。
- Line#.返回当前的行号，从每个新组的开始进行计数。例如：

  Master data

    1. Detail data
    2. Detail data
    3. Detail data

  Master data

    1. Detail data
    2. Detail data

- LineThrough#.返回当前行号，从报表的开始进行计数。例如：

  Master data

    1. Detail data
    2. Detail data
    3. Detail data

  Master data

    4. Detail data
    5. Detail data

- Column#.返回交叉报表中的当前栏数。
- Page#.返回当前页码。
- TotalPages.返回已完成报表的总页数。要使用这个函数必须使用扫描两遍报表。
- DayOf(<date>).返回所给日期的天数(1..31)。
- MonthOf(<date>).返回所日期的月份。
- YearOf(<date>).返回所日期日期的年份。
- MessageBox(<text>, <caption>, <buttons_and_icons>).显示一个包含文本，标题和按钮的消息对话框。返回用户选择的值(mrOk, mrCancel, mrYes, mrNo)。对于<buttons_and_icons>参数有下列值：

按钮	图标
mb_Ok	mb_IconError
mb_OkCancel	mb_IconQuestion
mb_YesNo	mb_IconInformation
mb_YesNoCancel	mb_IconWarning

在报表生成期间可以使用的过程和函数

- CurY.返回下一个 band 打印的 Y 轴坐标。你可以为 CurY 设定一个值｜｜它会因此移动位

置。*要在点阵和毫米间进行转换，使用下面比率：18 点阵/5 毫米*。

- FreeSpace.按点阵返回页面上的剩余的空白区域。
- FinalPass.如果扫描两遍的报表，现在是运行最后一遍，则返回 True。
- PageHeight.按点阵返回减去页脚 Band 高度的页面高度。
- PageWidth.按点阵返回页面宽度。
- StopReport.中止报表生成。
- NewPage.开始新的页面。
- NewColumn.在多栏报表中开始新的栏。
- ShowBand(<band>).显示名为<band>的 Band。

对象的属性和方法

报表的报表可视对象都是 TfrView 的继承类。在脚本中可以使用下列属性和方法：

属性	类型	描　　述
BandAlign	Integer	设置对象在 band 中的位置。 可选值：baNone, baLeft, baRight, baCenter, baWidth, baBottom。
Enabled	Boolean	确定对象是否响应事件。可选值：True, False。
FillColor	Integer	对象的背景颜色。颜色一般设为常量 clXXX。
FrameColor	Integer	对象边框的颜色。
FrameStyle	Integer	设置边框的类型。可选值：psSolid, psDash, psDot, psDashDot, psDashDotDot, psDouble。
FrameTyp	Integer	对象边框的类型，包含 frftTop, frftBottom, frftLeft, frftRight 的集合。
FrameWidth	Double	边框的宽度。
Height	Integer	所选对象的高度。
Left	Integer	以点阵确定对象在窗体上水平左边的位置。
Memo	String	所选对象的 memo 中的文本。这个属性可以通过它的索引来访问：例如：Memo[1]。
Memo.Count	Integer	返回 memo 中的总行数。
Name	String	所选对象的名称。
Stretched	Boolean	设置对象的尺寸和图形是否自动伸展。
Top	Integer	设置对象的顶边在它父对象的 y 轴坐标。
Visible	Boolean	确定对象是否在屏幕上显示。可选值：True 或 False。
Width	Integer	设置对象的宽度（水平尺寸）。

方法：

方法	类型	描　　述
Hide	-	通过设置 visiable 属性为 False 来隐藏对象。
Memo.Add	String	增加一个新行到 memo 的列表中。
Memo.Clear	-	删除一个对象的所有文本。

Memo.Delete	Integer	按索引参数删除指定的行。
Show	-	通过设置 visiable 属性为 True 来显示对象。

标准对象

u 对象"Text"(TfrMemoView)

除了上面的通用属性和方法，对象还有它自己的属性：

属性	类型	描 述
Alignment	Integer	指定文本在对象中的位置。可选值：frtaLeft, frtaRight, frtaCenter, frtaVertical, frtaMiddle, frtaDown。
AutoWidth	Boolean	确定对象是否根据对象中的文本自动改变大小。
CharSpacing	Integer	确定字符间距。
Font.Name	String	设置对象中文本属性的字体名称。
Font.Size	Integer	字体点阵大小。
Font.Style	Integer	设置字体修饰：加粗，斜体，下划线。可选值： fsBold, fsItalic, fsUnderline。
Font.Color	Integer	设置字体的颜色。
GapX	Integer	确定对象中文本和对象边框间的水平间距。
GapY	Integer	确定对象中文本和对象边框间的垂直间距。
HideZeros	Boolean	如果设为 True，那么变量为零值时隐藏。可选值：True; False。
LineSpacing	Integer	文本行间距。
Suppress	Boolean	Specifies whether or not repeat values are suppressed. 可选值：True; False。
TextOnly	Boolean	确定在生成过程中是否为变量。如果设为 True，那么变量不能生成。可选值： True; False。
WordBreak	Boolean	When a word wraps at the right margin, Wordbreak specifies whether wrapping occurs at the end of a syllable. Possible values: True; False (仅用于俄语).
WordWrap	Boolean	确定如果文本在对象中达到右边界时是否换行。可选值：True; False。

u 对象"Band" (TfrBandView)

属性	类型	描 述
Breaked	Boolean	Band 分隔开头。可选值：True; False。
ChildBand	String	Band 来自其它 band。
ColumnGap	Integer	对象中水平栏间距。
Columns	Integer	Band 中栏数。
ColumnWidth	Integer	栏宽。
Condition	String	指定分组条件。用于 band 的分组头。
DataSource	String	确定对象用于显示的数据源。
EOF	Boolean	判断是否已到数据集末尾。可选值：True; False。
FormNewPage	Boolean	在打印这个 band 和所有的细 band 时强制分页。可选值：True; False。
Master	String	指定 band 用于分组的数据。
OnFirstPage	Boolean	打印第一页。可选值：True; False。
OnLastPage	Boolean	打印最后一页。可选值：True; False。
PrintChildIfInvisible	Boolean	指定如果子 band 不可见时是否打印。可选值：True; False。
PrintIfSubsetEmpty	Boolean	指定如果子 band 为空时是否打印。可选值：True; False。
RepeatHeader	Boolean	指定是否在每页是重复这个 band。可选值：True; False.

方法：

方法	类型	描　　　　述
First	-	Band 依据的数据源移到第一条记录。
Next	-	Band 依据的数据源移到下一条记录。
Prior	-	Band 依据的数据源移到上一条记录。

- **对象 "Picture" (TfrPictureView)**

属性	类型	描　　　　述
BlobType	Integer	指定 Blob 字段包含图像的类型。 可选值：btBMP, btJPG, btICO, btWMF。
Center	Boolean	指定图像是否在对象上居中。 可选值：True; False。
DataField	String	指定包含图像的数据表字段。
KeepAspect	Boolean	指定当改变图像大小时是否保留比例关系。可选值：True; False。

附加对象

- **对象"Bar code" (TfrBarcodeView)**

属性	类型	描 述
DataField	String	包含数据的数据表字段。

- **对象"CheckBox" (TfrCheckBoxView)**

属性	类型	描 述
CheckColor	Integer	指定检查框选中时叉号的颜色。
CheckStyle	Integer	指定检查框的类型。可选值：csCross, csCheck。
DataField	String	包含数据的字段名。

- **对象"RichText", "RichText 2.0" (TfrRichView, TfrRXRichView)**

属性	类型	描 述
GapX	Integer	对象边框和对象中文本的水平间距。
GapY	Integer	对象边框和对象中文本的垂直间距。
TextOnly	Boolean	确定在生成过程中是否为变量。如果设为 True，那么变量不能生成。可选值：True; False。
DataField	String	包含数据的 blob 字段名。

- **对象"Rectangle with a shadow" (TfrRoundRectView)**

这个对象继承自"Text"对象，它拥有与父对象相同的属性和方法集，再加上下面的：

属性	类型	描 述
BeginColor	Integer	指定过渡色的初始颜色。
EndColor	Integer	指定过渡色的结束颜色。
Gradient	Boolean	指定是否使用过渡色。可选值：True; False。
RoundRect	Boolean	指定边框是圆角还是直角。可选值：True; False。
RoundSize	Integer	如果是圆角矩形，这个属性设置曲度。
ShadowColor	Integer	阴影颜色。
ShadowWidth	Integer	阴影宽度。
Style	Integer	如果使用过渡色，这个属性决定过渡类型。可选值：gsVertical, gsHorizontal, gsElliptic, gsRectangle, gsHorizCenter, gsVertCenter。

- **对象 "Shape" (TfrShapeView)**

属性	类型	描 述
Shape	Integer	指定一个可视的图形对象。 可选值： skRectangle, skRoundRectangle, skEllipse, skTriangle, skDiagonal1, skDiagonal2。

对话框控件

所有的对话框控件都继承自 TfrStdControl 类，并拥有下列属性和方法集：

属性	类型	描　　述
Color	Integer	对象的背景色。颜色可以通过常量 clXXX 指定。
Enabled	Boolean	确定对象是否响应事件。可选值：True, False。
Font.Name	String	设置对象中文本属性的字体名称。
Font.Size	Integer	字体点阵大小。
Font.Style	Integer	设置字体修饰：加粗，斜体，下划线。可选值： fsBold, fsItalic, fsUnderline。
Font.Color	Integer	设置字体的颜色。
Height	Integer	对象高度。
Left	Integer	以点阵确定对象在窗体上水平左边的位置。
Name	String	对象名。
Top	Integer	设置对象的顶边在它父对象的 y 轴坐标。
Visible	Boolean	确定对象是否在屏幕上显示。可选值：True 或 False。
Width	Integer	设置对象的宽度（水平尺寸）。

方法：

方法	类型	描　　述
Hide	-	通过设置 visiable 属性为 False 来隐藏对象。
SetFocus	-	设置输入焦点到这个控件。
Show	-	通过设置 visiable 属性为 True 来显示对象。

u　对象 "Label"

属性	类型	描　　述
Alignment	Integer	对象中文本行的对齐方式。 可选值： taLeftJustify, taRightJustify, taCenter。
AutoSize	Boolean	设置对象是否根据 label 文本宽度自动改变大小。 可选值：True; False。
Caption	String	标题是显示在 label 上的字符串。
WordWrap	Boolean	确定如果文本在对象中达到右边界时是否换行。如果为 True，则 AutoSize 属性为 False。可选值：True; False。

u　对象 "Edit"

属性	类型	描　　述
ReadOnly	Boolean	指定用户是否可改变编辑框中的内容。如果设为 True，那么值不不可以修改的。可选值：True; False。
Text	String	指定出现在编辑框中的文本。

u　对象 "Memo"

属性	类型	描　　述
Lines	String	指定 memo 对象中的文本行。单独行可以通过行号索引访问。Memo1. Lines [0].
Lines.Count	Integer	Memo 中的文本行号。
ReadOnly	Boolean	指定用户是否可改变 memo 中的内容。如果设为 True，那么值不不

		可以修改的。可选值：True; False。
Text	String	指定出现在 memo 对象中的文本。它在一行中包含的行。（使用 CR+LF 表示换行）

方法：

方法	类型	描述
Lines.Add	String	在字符串列表中增加一行。
Lines.Clear	-	在字符串列表中删除所有行。
Lines.Delete	Integer	在字符串列表中删除一行。这一行按指定的索引参数删除。

u 对象 **"Button"**

属性	类型	描述
Caption	String	标题属性是出现在按钮上的文本。
ModalResult	Integer	当用户在对话框中按这个按钮时，设置 ModalResult 用于关闭对话框。ModalResult 指定的值在调用 ModalResult 函数显示对话框后返回。 可选值：mrNone, mrOk, mrCancel。

u 对象 **"CheckBox"**

属性	类型	描述
Alignment	Integer	文本相对于选择框的对齐方式。 可选值：taLeftJustify, taRightJustify。
Caption	String	在复选按钮中显示标题。
Checked	Boolean	指定复选按钮是否选中。 可选值：True; False。

u 对象 **"RadioButton"**

属性	类型	描述
Alignment	Integer	文本相对于选择框的对齐方式。 可选值：taLeftJustify, taRightJustify。
Caption	String	在单选按钮中显示标题。
Checked	Boolean	指定单选按钮是否选中。 可选值：True; False。

u 对象 **"ListBox"**

属性	类型	描述
Items	String	在下拉列表中用于显示的项目数组。单独的行可以通过项目的索引来访问：ListBox1. Items [0]。
ItemIndex	Integer	选中行的索引号。
Items.Count	Integer	下拉列表中显示的数组行数。

方法：

方法	类型	描述
Items.Add	String	在列表框的字符串列表中增加一行。
Items.Clear	-	在列表框的字符串列表中删除所有行。
Items.Delete	Integer	在列表框的字符串列表中删除一行。这一行按指定的索引参数删除。

u 对象 **"ComboBox"**

属性	类型	描　　述
Items	String	在下拉列表中用于显示的项目数组。单独的行可以通过项目的索引来访问：ComboBox1. Items [0]。
ItemIndex	Integer	选中行的索引号。
Items.Count	Integer	下拉列表中显示的数组行数。
Style	Integer	类型属性决定下拉列表怎样显示它的项目。可以是带一个可以输入文本的编辑框的下拉列表(csDropdown) 也可以是不可以输入文本的编辑框的下拉下列列表(csDropwdownlist)，因此不可以编辑项目；或者是一个来自其它数据源的列表。 可选值：csDropDown, csDropDownList, csLookup。
Text	String	指定在下拉列表中出现的选中文本。

方法：

方法	类型	描　　述
Items.Add	String	在下拉列表的字符串列表中增加一行。
Items.Clear	-	在下拉列表的字符串列表中删除所有行。
Items.Delete	Integer	在下拉列表的字符串列表中删除一行。这一行按指定的索引参数删除。

数据访问控件

u 对象 "BDELookupComboBox"

这个对象继承自 TfrStdControl 类，除了拥有相同的基本属性和方法集外，还有下列扩展属性和方法集：

属性	类型	描 述
KeyField	String	字段，选择值的依据。
ListField	String	在列表中显示值的字段。
ListSource	String	数据源。
Text	String	选中的值。

u 对象 "BDETable"

属性	类型	描 述
Active	Boolean	确定数据表是否激活。
DatabaseName	String	数据库别名。
Fields	Variant	可用的字段列表。属性可以通过索引，字段名访问：a: = Query1. Fields ['Customer']。
FieldCount	Integer	数据表的字段数。
Filter	String	记录过滤表达式。
IndexName	String	次要索引名。
MasterFields	String	用于联接主数据集的字段。
MasterSource	String	主数据集。
TableName	String	数据表名。
EOF	Boolean	如果到了记录集的末尾，为 True。
RecordCount	Integer	数据表中的记录数。

方法：

方法	类型	描 述
Open	-	打开数据表。等同于 Active: = True。
Close	-	关闭数据表。等同于 Active: = False。
First	-	按索引在数据表中移到第一条记录。
Last	-	按索引在数据表中移到最后一条记录。
Next	-	按索引在数据表中移到下一条记录。
Prior	-	按索引在数据表中移到上一条记录。

u 对象"BDEQuery"

属性	类型	描 述
Active	Boolean	确定查询是否激活。
DatabaseName	String	数据库别名。
DataSource	String	主数据集。
Fields	Variant	可用的字段列表。属性可以通过索引，字段名访问：a: = Query1. Fields ['Customer']。
FieldCount	Integer	数据集的字段数。
Filter	String	记录过滤的表达式。
SQL	String	查询语句。属性可以通过索引访问：Query1. SQL [0]。
SQL.Count	Integer	查询语句的行数。

EOF	Boolean	如果到了记录集的末尾，为 True。
RecordCount	Integer	数据表中的记录数。

方法：与 BDETable 相同，并加了一些扩展：

方法	类型	描 述
SQL.Add	String	增加一行。
SQL.Clear	-	清除所有行。
SQL.Delete	Integer	按索引删除指定的行。

u 对象 "BDEDataBase"

属性	类型	描 述
AliasName	String	选择用于连接到数据库的别名。
Connected	Boolean	如果为 True，激活连接。
DatabaseName	String	加到别名列表中的名字。
DriverName	String	连接到数据库的驱动程序名。
LoginPrompt	Boolean	定义是否要求用户输入数据库连接口令。
Params	String	连接参数。属性可以通过索引访问：DataBase1. Params [0]。
Params.Count	Integer	参数列表总行数。

方法：

方法	类型	描 述
Params.Add	String	增加一行。
Params.Clear	-	清除所有行。
Params.Delete	Integer	按索引删除指定的行。

解释器的使用

下面是一些使用解释器的例子：

1、要按顺序加亮求和项：白色，表示合计小于 2000；绿色，表示合计在 2000 与 1000 之间；红色，表示合计在 1000 之内，在求和对象中输入下面脚本：

```
if [Summa] < 2000 then
  FillColor := clTransparent
else if [Summa] < 10000 then
  FillColor := clGreen
else
  FillColor := clRed
```

[Summa]是需要求和的变量或字段。你也可以使用数值常量来选取颜色：

```
FillColor := 128 + 128*256 + 128*65536 //(gray color)
```

2、要只显示合计大于 2000 的数据行，在相关数据 band 的脚本中加入下面代码（你可以在 memo 编辑器中按 Ctrl+Enter 显示它）：

```
if [Summa] > 2000 then
  Visible := 1 else
  Visible := 0
```

3、使用 FreeSpace 函数确定在当前页中有多少空白区域。如果不够，调用 NewPage 过程插入“手工分页符”并开始新一页。

```
if FreeSpace * 5/18 < 30 then NewPage
```

4．使用 CurY 属性移动当前坐标。例如，要放置报表合计 band 到页面的底部，在报表合计中使用这段代码：

```
CurY := PageHeight - Height
```